

Lernen mit AIM, Teil I

Lernen mit dem AIM

Eine "AIM-Einführung", Teil 1

Einleitung

Seit etwa einem Jahr ist er auf dem internationalen Microcomputer-Markt erhältlich: Der AIM 65 aus dem Haus des amerikanischen Halbleiterherstellers ROCKWELL INTERNATIONAL. Für knapp unter DM 1.000,- wird dieses Kleinsystem in seiner Grundversion angeboten, mit einer schreibmaschinenähnlichen ASCII-Tastatur, einem Thermodrucker und einem alphanumerischen Anzeigendisplay, allerdings ohne Gehäuse und Netzteil.

Als ich vor gut einem dreiviertel Jahr, bisher an das Arbeiten mit einem SC/MP-System gewöhnt, nach einem modernen und komfortableren Gerät Ausschau hielt und diesen Einplatinencomputer kennenlernte, wußte ich sofort, welches Gerät für mich das passende ist! Im Gegensatz zu den fertigen Hobby-Computern wie TRS-80, PET, APPLE u. a. ist der AIM nicht primär für die Programmierung in der relativ einfachen höheren Programmiersprache BASIC konzipiert (obwohl er freie Steckplätze für ein 8K-BASIC auf der Hauptplatine hat), nein, seine Stärke liegt in der leistungsfähigen Programmierung auf Maschinenebene. In jeder Hinsicht ist der AIM 65 ein "offener" Microcomputer. Dies bedeutet hier nicht nur hardwaremäßige Offenheit durch übersichtlich und leicht zugänglich angeordnete Bauteile und die Anschlußmöglichkeiten für 2! Cassettenrecorder und amerikanischen TTY-Fernschreiber sowie die universellen Ein-/Ausgabemöglichkeiten durch den hochintegrierten VIA6522 Baustein, mit Offenheit ist hier auch eine softwaremäßige Offenheit gemeint.

Alles ist genau dokumentiert, von Haus aus wird der AIM mit Anwenderhandbuch, Monitor-Programm-Listing, Programing-Manual und Hardware-Manual geliefert, und die vielen Programmbeispiele fordern förmlich dazu auf, selbst ein Maschinenprogramm zu entwickeln.

Das System weist alle Merkmale eines kleinen Entwicklungssystems auf, wie sie sonst nur bei

wesentlich teureren Geräten der Computerhersteller für die Programmentwicklung zu finden sind: 8K Monitorprogramm mit Textverarbeitungsmöglichkeiten (Editor), Mini-Assembler und Disassembler, zahlreichen Debugging-Hilfen u. a. Ein professioneller 4K Assembler ist als ROM-Baustein zusätzlich erhältlich. Die CPU des Gerätes ist der 6502 Microprozessorbaustein, wie er in den meisten Kleincomputersystemen zu finden ist und wie er manchem Leser vom KIM her vertraut sein wird.

Überhaupt schließt sich der AIM in vielen Punkten an den KIM an, so etwa bei seiner Fähigkeit, die KIM-Cassettendaten mit der wesentlich langsameren KIM-Baudrate zu laden.

Dieser Beitrag soll der Anfang einer Artikelserie sein, die sich nur mit dem AIM beschäftigt. Entsprechend den Möglichkeiten und Stärken des AIMS werden wir uns vorerst nur mit der bytenahen Maschinenprogrammierung befassen und dabei meist das Zusammenspiel des Gerätes mit unterschiedlichen Hardwarezusätzen im Auge haben. Es wird uns weniger auf perfekte Lösungen ankommen als darauf, mit dem Gerät zu lernen und Freude an der eigenen Arbeit zu haben. Zu jedem Artikel der Serie werden einige Aufgaben und Möglichkeiten angeboten, wie der Leser die vorgestellten Beispiele selbst durch eigene Arbeit vervollständigen kann. Es ist allerdings überhaupt nicht möglich, auch nur annähernd alle Möglichkeiten des AIMS in einer solchen Serie voll auszuschöpfen. Deshalb werden wir immer exemplarisch und nahe am praktischen Beispiel bleiben. Im einzelnen sind folgende Beiträge vorgesehen:

- Aufbau und Anschluß eines Netztesiles
- Anschluß eines Cassettenrecorders an den AIM
- Uhrenprogramm
- Analog/Digital- und Digital/Analogwandlung
- Meßwerterfassung und Ausdruck unter Zuhilfenahme des A/D-Wandlers und des Uhrenprogrammes

- Anschluß eines Heathkit-Druckers
- Hexadezimal in Dezimalumwandlung
- Dezimal in Hexadezimalumwandlung
- Musikerzeugung
- Steuern und Zählen mit dem komplexen E/A-Baustein
- Anzeige analoger Daten mit dem Oszilloskop
- Spiele u. a.

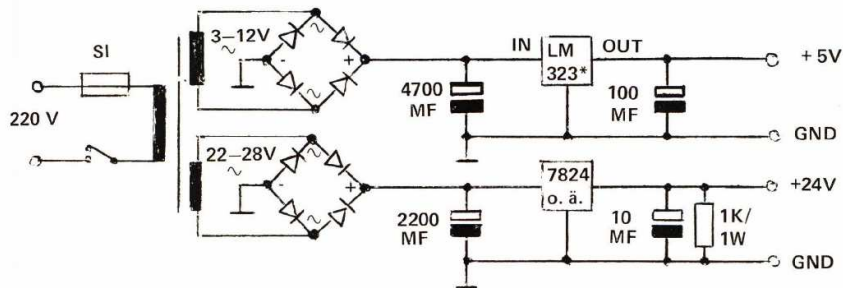
1.1 Brauchen Sie ein Netzteil?

Wenn Sie kein Netzteil brauchen und Ihr Gerät schon in Betrieb ist, so können Sie die folgenden Abschnitte getrost überspringen.

Bevor wir mit dem AIM arbeiten können, müssen wir das Gerät aus seinem Transportkarton auspacken, die Tastaturplatine mit der Hauptplatine verbinden. Druckerpapier einlegen und die Spannungsversorgung anschließen. Am besten läßt sich die Zeit, die zwischen Bestellung des Gerätes und Auslieferung vergeht, dadurch nutzen, daß man ein Netzteil selber baut, wenn man nicht die rund DM 300,- ausgeben will,

die eine fertige Spannungsversorgung kostet. Der Bau eines Netzteiles hingegen kostet unter DM 100,- und ist nicht weiter schwierig.

Obwohl die Anschlußleiste der Spannungsversorgung am Rechner für vier Spannungen ($\pm 12V$, $+5V$, $+12V$ u. $+24V$) vorgesehen ist, so brauchen wir vorerst nur die $+5V$ und $+24V$ Versorgung, solange wir kein TTY anschließen wollen. Die $+5V$ sollten 2 Ampere liefern können, die 24 Volt Versorgung kommt mit weniger aus. Es sind hier vor allem kurze Spitzenbelastungen, die mit einem geeigneten Ladeelko aufgefangen werden können. Diese kurzen Belastungsspitzen werden durch den Drucker erzeugt, und nur für diesen Drucker ist die 24Volt-Versorgung überhaupt nötig. Außerdem ist es nicht nötig, daß die Druckerspannung stabilisiert ist. Der Einfachheit halber nehmen wir aber auch hier, ebenso wie bei der 5V Spannungsstabilisierung, einen integrierten Festspannungsregler. Die Schaltung unseres Netzteiles zeigt Bild 1.



*) siehe Text

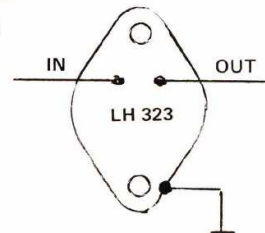


Bild 1

Statt des integrierten Festspannungsreglers LM323 lassen sich auch zwei vom Typ LM 309 o. ä. verwenden, die wir einfach parallel schalten können. Das funktioniert ausgezeichnet, diese Lösung wurde bereits zig-fach ausprobiert, es geht!

Folgende Punkte sind beim Aufbau des Netzteiles wichtig, der mit Hardware und Lötkolben noch ungeübte Leser möchte diese noch bitte beachten:

1. Die Festspannungsregler müssen auf Kühlkörper montiert werden. Besonders bei der 5V Versorgung nach dem Motto handeln: "Viel hilft viel", und diesen Kühlkörper besser eine Nummer zu groß als zu klein wählen. Für den 24V Regler ist das nicht so kritisch, hier reicht ein kleiner Kühlkörper.

2. 5V und 24 Volt Regler benötigen entweder getrennte Kühlkörper, oder es muß eine Isolierung zwischen den Bauteilen und dem Kühlkörper vorgesehen werden.
3. Die Leitungen möglichst kurz und dick gestalten, es fließen mittelmäßig starke Ströme.
4. **Der wichtigste Punkt:** Zum Schutz der Gesundheit von Programmierer und Rechner muß der Transformator völlig isoliert werden. Man baue am besten das ganze Netzteil in ein billiges Plastikgehäuse ein.

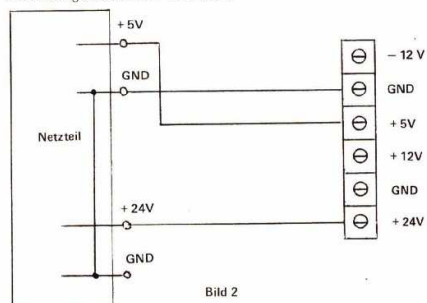
Zum Aufbau des Gerätes eignet sich ein Lötösenstreifen oder eine Lochrasterplatte, auf der die Bauteile verlötet werden können. Wenn Sie diese Punkte beachten, dann wird sich der Aufbau Ihres Netzteiles problemlos gestalten und Sie sparen viel Geld.

1.2 Anschluß des Netzteiles

Nachdem die Teile des AIMS aus dem Karton entnommen wurden, sollte zuerst überprüft werden, ob sich Reste des Verpackungsmaterials in den mechanischen Teilen des Rechners befinden. Falls Ihr Gerät in der Grundversion geliefert wurde und Sie die RAM-Speichererweiterung noch einsetzen müssen, so ist darauf zu achten, daß die MOS-RAMs beim Einsetzen nicht durch statische Elektrizität zerstört werden. Es empfiehlt sich, wenn man einen Baustein einsetzt, zuvor stets über die Anschlußpunkte an der Rechnerplatine zu streifen, während man den Baustein mit samt der Verpackung in der Hand hält. Diese Vorsichtsmaßnahme zur Ableitung statischer Aufladung reicht wohl aus, denn mir ist bisher (auf diese Art) noch kein Baustein zerstört worden. Beim Einsetzen der IC's Ruhe bewahren, darauf achten, daß der Baustein richtig in der Fassung sitzt (die Einkerbung im Speichergehäuse muß mit dem Aufdruck auf der AIM-Platine übereinstimmen) und kein Schaumstoff zwischen den IC-Anschlüssen hängengeblieben ist.

Im Verpackungskarton befinden sich kleine Plastikknoppen, die klebt man möglichst gleichmäßig verteilt unter die Platinen von Rechner und Tastatur. Als nächstes müssen diese beiden Platinen mit dem bunten Verbindungskabel elektrisch verbunden werden. Diese Sockelstecker vom Verbindungskabel sind aus ziemlich hartem Metall, wenn beim Einstecken ein Stift versehentlich verbogen wurde, so kann es leicht passieren, daß er abbricht. Hier ist also Vorsicht geboten.

Legen Sie jetzt das Druckerpapier ein. Solange kein Papier im Drucker ist, darf dieser nicht in Betrieb genommen werden.



Beim Einlegen muß das Papier so eingeführt werden, daß es nach unten (zur Platine) von der Rolle abrollt. Um den Rechner in Betrieb zu nehmen, muß nur noch die Spannungsversorgung angeschlossen werden. Klemmen Sie zuerst die 5 Volt- und die Masseleitung an die Schraubklemmen links oben auf der Platine an, wie in Bild 2 dargestellt. Nach Einschalten der Spannung müßte zuerst die Leuchtschrift ROCKWELL AIM 65 auf dem Display erscheinen und dann die Schrift PRINTER DOWN. Wenn nichts dergleichen geschieht, überprüfen Sie folgende Punkte:

- Ist das Netzteil in Ordnung? Sind die Anschlüsse der + 5V - Versorgung richtig an den Schraubklemmen angeschraubt, ohne daß sich die Drahtisolierung zwischen Draht und Klemme befindet?
- Stehen die beiden kleinen Schiebeschalter links auf der Rechnerplatine auf Stellung "RUN" und "KB"?
- Befinden sich eventuell elektrisch leitende Drahtenden o. ä. unter der Platine?

Drücken Sie die Reset-Taste. Passiert immer noch nichts, so liegt eindeutig ein Defekt vor.

1.3 Inbetriebnahme des Druckers

Ist bis jetzt alles in Ordnung, so sollten Sie nun die Druckerspannung anschließen und den Drucker in Gang setzen. Schließen Sie die 24V-Leitung an. Wenn in Ihrem Netzteil die Masseleitungen von +5V und +24V miteinander verbunden sind, so brauchen Sie für die 24 Volt-Versorgung keine extra Masse anschließen, denn auf der Rechnerplatine sind die Masseleitungen ebenfalls zusammengelegt.

1.4 Druckerbedienung

Alle Arbeitsschritte, die Sie zum Eingeben und Protokollieren von Programmen etc. vollziehen, werden vom Display angezeigt. **Wahl-**

weise wird der angezeigte Text auch vom Drucker ausgegeben. Die Druckerausgabe können Sie ein- und ausschalten, indem Sie erst die Taste CTRL und dann gleichzeitig die Taste PRINT drücken. Je nachdem, ob der Drucker nun ein- oder ausgeschaltet ist, erscheint ON bzw. OFF auf dem Display. Unabhängig vom Betriebszustand des Druckers können Sie die aktuelle Displayanzeige aber auch durch einfaches Drücken der Taste PRINT ausdrucken lassen.

Wenn Sie einen Papiervorschub ohne Druckvorgang wollen, dann können Sie dies durch Drücken von LF erreichen.

Übrigens: Der Drucker ist relativ schnell — In einer Sekunde druckt dieser Thermodrucker etwa zwei Zeilen à 20 Zeichen.

Wird fortgesetzt.

M. Krüger, Berlin ■

Preiswerte Normalpapierdrucker für 120mm- und DIN A4-Rollen

- X50 — 52 Zeichen pro Zeile, 120 mm-Papier
- X88 — 40 Zeichen oder softumschaltbar; 80 Zeichen pro Zeile, 120 mm-Papier
- X80 — 80 Zeichen pro Zeile, 210 mm-Papier (DIN A4)
- PET/CBM-Anschlußfertig mit IEC-Kabel
- Voll adressierbar



PREISE: X50/X88 DM 1808,— (inkl. MwSt.)
X80 DM 2147,— (inkl. MwSt.)

- Alle Commodore-Sonderzeichen plus Cursor-Symbole
- Umschaltbar für Commodore 2001- oder 3001-Zeichenvorrat
- V24 und 20 mA mit ASCII-Zeichenvorrat lieferbar

adcomp

Datenysteme GmbH · Nordendstraße 8
8000 München 19 · Telefon 089/19 40 19

COREX COREX GmbH
Handelsagentur GmbH Feldstr. 25
3680 Bad Homburg
Tel.: 06172/ 4 42 08

stellt vor:

Die Druckersensation

80 Column Impact Drucker

(72, 80, 96, 120 oder 132 Zeichen wahlweise pro Zeile unter Softwarekontrolle).

Ideal für Industrieinsatz. Metallgehäuse, Zeichensatz vom Anwender tlw. definierbar. Bis zu sechs verschiedene Zeichensätze können unter Programmkontrolle gewählt werden. Zwei Sätze sind im Standarddrucker enthalten.

Zeichengröße: 5 x 7 Punktmatrix
ASCII Standard-Zeichensatz

Breitschrift

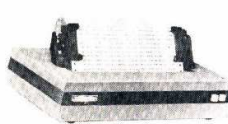
15 Baudraten zwischen 75 und 19200 Baud.
RS232, 20mA, IEEE 488 und Centronics Schnittstelle. Hohe Zuverlässigkeit.

Optionen: 2K Buffer, Tractor Feed.

Preis DM 1.977,— incl. MwSt.

Doppel floppy für PET (800K)

Betriebssystem Computhink . . DM 3.486,— inkl.



SPIMA Computer GmbH
Turbinenstr. 4
6800 Mannheim 31
Tel.:
0 62 21/ 72 15 15
Telex: 463 708 spima d

TI 99/4

Es ist eine Tatsache, daß der neue TI 99/4 einer der meistgeschätztesten Home-Computer heutzutage ist. Da die Nachfrage die Liefermöglichkeiten z. Zt. weit übersteigt und wir versuchen wollen, allen Kunden gerecht zu werden, wurde nachfolgende Abwicklung vorgesehen:

Und so funktioniert's:
Sie senden uns Ihre Bestellung für einen TI 99/4 mit einer Anzahlung von DM 500,— per Verrechnungsscheck. Nach Erhalt Ihrer Bestellung und Ihrer Anzahlung wird diese mit dem Eingangsdatum auf unsere Prioritätenliste eingetragen und wir senden Ihnen eine Empfangsbestätigung und Ihre Position auf unserer Prioritätenliste zu.

So, wie wir unsere Zuteilung der TI 99/4 erhalten, werden wir dieselben an unsere Kunden in der Reihenfolge der Prioritätenliste ausliefern. Mit einem Wort: First in, First out!

Garantie:

Sie können Ihre Bestellung jederzeit innerhalb von 14 Tagen nach Auftragserteilung widerrufen und erhalten Ihre Anzahlung in Höhe von DM 500,— wieder voll zurück. Sie können uns jederzeit anrufen. Wir geben Ihnen dann Auskunft über den Stand Ihrer Bestellung auf unserer Prioritätenliste und wann Sie voraussichtlich mit Ihrer Lieferung rechnen können.



Bitte wenden Sie mir/juns folgende Produkte:

TI Homecomputer Produkt:	
PHC 004 M TI 99/4 Home Comp.	
einschl. Farbmonitor	DM 3380,—
Anzahl.	DM 500,—
PHA 2000 Dual-Cassette Kabel	DM 25,—
PHP 1100 Remote Controls (Paar)	DM 99,—
PHP 1500 Sprach Synthesizer	DM 495,—
Module	
PHM 3000 Diagnose	DM 99,—
PHM 3001 Demonstrations	DM 210,—
PHM 3002 Early-Learning Fun	DM 99,—
PHM 3003 Beginning Grammar	DM 99,—
PHM 3004 Number Magic	DM 66,—
PHM 3005 Farb-Graphic	DM 66,—
PHM 3006 Home-Func. Desktop	DM 99,—
PHM 3007 Househ. Budget Plan	DM 149 50,—
PHM 3008 Video Chess (Schach)	DM 210,—
PHM 3009 Football	DM 99,—
PHM 3010 Physical Fitness	DM 99,—
Endsumme	

Die Preise beinhalten 12% MwSt. + Versandkosten, wahlw. (DM 45,20 für Computer einschl. Farbmonitor, DM 3,90 für Module etc.)

Beiliegend Scheck in Höhe von DM.

Name:

Strasse:

PLZ/Ort:

Land:

Telefon:

Datum:

Unterschrift:

(Bitte Monogramme, Unterschriften d. Anzahlungsberechtigten)

ELCOMP 7

Lernen mit AIM, Teil II

Lernen mit dem AIM

Eine "AIM-Einführung", Teil 2

Im ersten Teil der Artikelserie wurden alle notwendigen Schritte besprochen, um den AIM zu einem funktionsbereiten Gerät zu machen. Dem Leser wurde ein Weg gezeigt, wie er mit wenig Aufwand an Arbeit und Geld ein passendes Netzteil, welches genau auf den AIM zugeschnitten ist, aufbauen und in Betrieb nehmen kann.

Im zweiten Teil wird es uns um die ersten "Gehversuche" mit dem AIM gehen. Nachdem die Grundzüge der Bedienung des Gerätes mit den für uns notwendigen Monitorbefehlen erläutert wurden, sollen einige wichtige und für uns sehr nützliche Subroutinen aus diesem im Gerät fest abgespeicherten Monitorprogramm besprochen werden. Besonders nützlich waren für meine eigene Arbeit vor allem die Routinen für die Dateneingabe über die Tastatur und die Datenausgabe mittels Thermodrucker und Anzeigedisplays. Weiterhin wird ein (nicht nur) für den AIM kompatibler Hardwarezusatz zur Musikerzeugung vorgestellt, wie er eigentlich an keinem Lern- oder Hobbycomputer fehlen dürfte! Mit diesem Zusatz hat der Leser die Möglichkeit, fast spielerisch sein Gerät genauer kennenzulernen. Die Ansteuerung der Musikhardware erfolgt über einen Port des hochkomplexen Ein-/Ausgabebausteins VIA 6522, der ja im AIM dem Anwender voll zur Verfügung steht. Bei der Besprechung der Software zur Musikerzeugung können wir uns also mit diesem VIA gleich ein wenig vertraut machen.

Die Grundzüge der Programmierung des Prozessors 6502 werden nicht detailliert behandelt. Nur in besonderen

Fällen, wo es um einen Programmiertrick oder einen ungebrauchlichen Befehl geht, werden wir dies erläutern. Die Beschreibung von Registerstruktur, Befehlstabelle und Adressierungsarten des 6502 würde ein Handbuch für sich alleine füllen. Solche Bücher sind ja bereits geschrieben worden.

2.1 Die Eingabe eines Programms

Um ein Programm zu erstellen, brauchen wir uns nicht, wie bei Microcomputersystemen der ersten Generation üblich, um die binäre, oktale oder hexadezimale Darstellung der Maschinenbefehle kümmern. Der AIM 65 hat ja von Haus aus im Monitorprogramm einen Klein-Assembler, der die mnemonische Befehlseingabe ermöglicht. Wir schalten den Rechner ein und drücken:



Der AIM zeigt den aktuellen Stand des Programmzählers an, und wartet nun auf die Eingabe eines Programms. Wir können den Befehlszählerstand ändern durch:



und Eingabe der hexadezimalen Adresse

an welcher unser Programm beginnen soll. So z. B.:

0200, nach RETURN wird der aktuelle Befehlszählerstand wiederum links auf dem Display angezeigt.

Wir geben folgendes Programm ein:

0200	LDA # 00	ACCU UND
0202	TAX	INDEX AUF 00 SETZEN
0203	STA 0300,X	INHALT VOM ACCU IN ADRESSE 0300+WERT AUS IND.X ABSPEICHERN
0206	INX	INDEX X UM 1 ERHÖHEN
0207	CPX # 00	IST X WIEDER 00?
0209	BNE 0203	WENN NICHT, DANN SPRUNG NACH 0203 UND ERNEUTER SCHLEIFENDURCHLAUF
020B	JMP E182	RÜCKSPRUNG ZUM MONITOR

Merke: Beim AIM-Assembler werden die Adressen der Branch-Befehle absolut eingegeben. Der Assembler rechnet selbstständig in die relative Sprungadresse um!

Dieses Programm füllt den Speicherbereich von 0300 bis 03FF mit 00. Der erfahrene 6502-Programmierer sieht sofort, daß in diesem Programm ein Befehl überflüssig ist. Suchen Sie diesen Befehl! Ein Tip: Es ist ein 2-Byte-Befehl.

Die Adresse E182 als Einsprung für den Monitor sollte man sich merken!

2.2 Start eines Programms

Nach der Eingabe des letzten Befehls drücken wir ESC, um wieder "in den Monitor zu kommen". Mit

SHIFT und Eingabe von 0200 setzen wir den

Programmzähler erneut auf den Programmanfang. Der Start des Programms erfolgt mit

G **RETURN**. Im selben Moment erscheint

wieder der Monitor-Prompt auf dem Display. Es ist verblüffend, wie schnell so ein Prozessor arbeitet! Unser Programm wurde im Bruchteil einer Sekunde ausgeführt.

Wie können wir nun prüfen, ob das kleine Programm korrekt ausgeführt wurde? Ganz einfach:

Wir drücken die Monitorbefehlstaste **M**. Der AIM fragt nach der Adresse: $\langle M \rangle =$

Wir geben 0300 ein und drücken RETURN.

Wurde das Programm richtig eingegeben, so werden die ersten vier Byte mit 00 angezeigt. Durch drücken der SPACE-Taste können wir jeweils die vier nachfolgenden Byte prüfen, diese müssen ebenfalls Null enthalten, bis wir bei 03FF angelangt sind.

2.3. Disassemblieren eines Programms

Das Programm wurde korrekt eingegeben und gestartet, seine Funktion wurde überprüft. Wie erlaubt der Monitor eine Auflistung des Programms? Mit dem K-Command benutzen wir den Disassembler. Wir drücken **K**. Der Computer "fragt":

$\langle K \rangle * =$ mittels des Sternchens, ab welcher

Adresse das Programm, was nun im RAM-Speicher steht, disassembliert und angezeigt werden soll. Mit RETURN wird nur ein Befehl disassembliert, mit SPACE oder . sind es alle folgenden.

Für unser Programmbeispiel ist 07 einzugeben. Je nachdem, ob der Drucker (Druckerbedienung siehe ELCOMP Nr. 3) aus- oder eingeschaltet war, wird unser disassembliertes Programm nur auf dem Display oder auf Drucker und Display ausgegeben.

```
0200 LDA # 00
0202 TAX
0203 STA 0300,X
0206 LDA # 03
0208 JSR EA46
```

2.4 Eigene Programmversuche

Die Bedienung des AIM ist an Hand der dargestellten Beispiele so weit verständlich geworden, daß der Leser in der Lage ist, einfache Übungsprogramme einzugeben bzw. zu verändern und auszugeben.

Bis man sich in alle Möglichkeiten des Monitors einge- arbeitet hat, vergeht eine gewisse Zeit. Der Monitor ist mit 8K Byte ja nicht gerade klein.

Nun sollte der Leser versuchen, selbst kurze Übungspro- gramme zu erstellen. Hier sollen ein paar Möglichkeiten angedeutet werden, wie unser unter 2.1 vorgestelltes Programm zur Löschung der Speicherbytes von 0300 – 03FF veränderbar wäre:

- 1.) In jedem Byte soll eine fortlaufende Nummer stehen:
In 0300 steht 00, in 0301 steht 01 usw.
- 2.) Versuchen Sie ein Programm zu erstellen, das mehr als 256 Byte löscht. Aber passen Sie auf: Überschreiben Sie nicht Ihr eigenes Programm oder eventuell benutzte Speicherstellen auf Page Zero oder im Stack!

Im nächsten Heft werden mögliche Lösungen für die gestellten Aufgaben vorgestellt.

2.5 Ausgabe auf Display und Drucker

Wollen wir mit dem Computersystem kommunizieren, so müssen wir in der Lage sein, Daten direkt per Tasta- tur einzugeben oder über Drucker und Display auszuge- ben.

In diesem Abschnitt werden zunächst einige der Aus- gaberoutinen besprochen. Im Anwenderhandbuch sind die wichtigsten Subroutinen aufgeführt.

- **NUMA ab EA46**
Diese Subroutine interpretiert den Akkuinhalt als zweistellige Hexadezimalzahl und gibt diese an Display und Drucker aus, bzw. nur an das Dis- play, in Abhängigkeit davon, ob der Drucker ein- geschaltet ist.
- **CRLOW ab EA13**
Diese Routine bewirkt, daß ein "Wagenrücklauf- zeichen" (CR=0D) und ein "Zeilenverschubzei- chen" (LF=0A) ausgegeben werden. Dies bewirkt praktisch, daß die nächste geschriebene Zeile un- ter die vorangegangene geschrieben wird.

Wir geben mittels des I-Commands folgende modifi- zierte Version unseres Speicherlöschprogramms ein:

```
NEUER PROGRAMMTEIL:
HEX 03 UMWANDELN UND AN DISPLAY
UND DRUCKER
```

Ø2ØB	TXA	INDEX X IN ACCU
Ø2ØC	JSR EA46	AUSGABE AN DISPLAY UND DRUCKER
Ø2ØF	JSR EA13	ZEILE VORSCHIEBEN
Ø212	LDA # ØØ	ØD IN ACCU
Ø214	INX	
Ø215	CPX # ØØ	
Ø217	BNE Ø2Ø3	
Ø219	JMP E182	

Experimentieren Sie mit den vorgestellten Subroutinen!
Folgendes Übungsprogramm bietet sich an:

- 1.) Programmieren Sie einen Zähler mit vier Displaystellen. (Eine Subroutine, die die Displayausgabe wesentlich schneller vollzieht, so daß das lästige Flackern der Displaydigits wegfällt, wird in einer späteren Ausgabe besprochen).
- **OUTALL ab E9BC**
Diese Subroutine interpretiert den AKKU-Inhalt des ASCII-Zeichens und gibt ihn an Display/Drucker aus, je nachdem, ob der Drucker eingeschaltet ist.
- **OUTDIS ab EF05**
Der AKKU-Inhalt wird als ASCII-Zeichen nur an das Display ausgegeben.

Zum Kennenlernen dieser beiden Ausgaberroutinen hier folgendes Programm:

Ø2ØØ JSR EA13	Ø21C LDA # 4D
Ø2Ø3 LDA # 2Ø	Ø21E JSR Ø22A
Ø2Ø5 JSR Ø22A	Ø221 LDA # 5Ø
Ø2Ø8 LDA # 45	Ø223 JSR Ø22A
Ø2ØA JSR Ø22A	Ø226 JMP E182
Ø2ØD LDA # 4C	Ø229 NOP
Ø2ØF JSR Ø22A	Ø22A JSR E9BC
Ø212 LDA # 43	Ø22D LDA # 2Ø
Ø214 JSR Ø22A	Ø22F JSR EFØ5
Ø217 LDA # 4F	Ø232 RTS
Ø219 JSR Ø22A	

Dieses Programm wurde bewußt nicht kommentiert, lassen Sie sich überraschen. Bitte versuchen Sie, den Ablauf dieses recht übersichtlichen Programms zu verstehen. Glauben Sie bitte nicht, daß es nicht besser und effektiver programmiert werden kann, als in diesem Beispiel! Die elegantere und kürzere Lösung sollen Sie selbst finden.

2.6 Tastaturroutinen

Zur Eingabe von Daten mittels der Tastatur stehen eine Reihe von Subroutinen zur Verfügung. Hier soll nur auf die gebräuchlichste näher eingegangen werden. Mit dieser Routine läßt sich ganz gut operieren:

- **READ ab E93C**
Ein Zeichen wird von der Tastatur gelesen und als ASCII-Wert im Akku behalten.

Das folgende Programm benutzt neben READ und den anderen bereits bekannten Routinen die Subroutine:

- **EQUAL ab E7D8**
Welche ein Gleichheitszeichen auf das Display schreibt.

Dieses Programm erlaubt die Umwandlung eines ASCII-Charakters, wie er mittels der Tastatur eingegeben wird, in die ihm zugehörige Dezimalzahl.

Ø2ØØ JSR E93C	READ
Ø2Ø3 JSR E9BC	OUTALL
Ø2Ø6 STA ØØ	ACCU IN ØØ
	ZWISCHENSPEICHERN
Ø2Ø8 JSR E7D8	EQUAL
Ø2ØB LDA ØØ	ACCU AUS ØØ LADEN
Ø2ØD JSR EA46	NUMA
Ø21Ø JSR E93C	READ

Ø213	STA	ØØ	ACCU ZWISCHENSPEICHERN
Ø215	JSR	EA13	CRLOW
Ø218	JMP	Ø2Ø3	RÜCKSPRUNG AN ANFANG

Neben der eben beschriebenen Routine READ gibt es eine Reihe anderer für die verschiedenen Anwendungsfälle nützliche Tastaturroutinen, die sich zum Teil nur durch kleine Differenzen in ihrer Wirkung voneinander unterscheiden. Es empfiehlt sich, die einzelnen Subroutinen in Experimentierbeispielen zu verwenden und untereinander zu vertauschen, um sie so besser kennenzulernen.

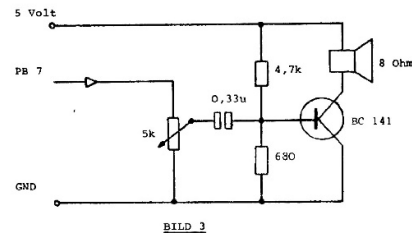
Die wichtigsten Routinen sind vor allem:

- CUREAD ab FE82
- RBYTE ab E3FD
- RCHEK ab EØ97
- RDRUB ab E95F
- RED1 ab FE96
- REDOUT ab E973

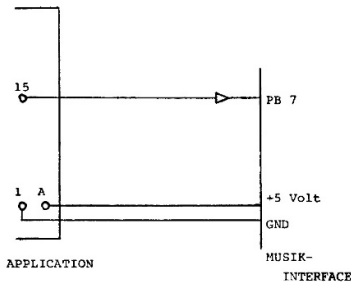
Ein Teil dieser Routinen sind kombinierte Ein-/Ausgeroutinen, die ebenfalls Display und/oder Drucker ansteuern können.

3.1 Wir machen Musik

Nach vielen trockenen Übungsbeispielen, wie wir sie bisher behandelt haben, soll nun etwas konkret Sinnliches im Mittelpunkt unseres Interesses stehen:



In Bild 3 ist eine kleine Schaltung zu sehen, mit der wir unseren Rechner in ein Experimentiermusikinstrument verwandeln können. Der Aufbau dieser Schaltung ist wirklich ganz einfach. Als Lautsprecher ist ein 8 Ohm-Typ vorgesehen, wie er in den billigen japanischen Transistorradios verwandt wird und in jedem Elektronikzubehörhandel zu kaufen ist. Das Potentiometer wurde deshalb vorgesehen, damit unsere "Musik" auch einmal leise bzw. abzustellen ist, während ein Programm läuft – allzuleicht handelt man sich sonst den Zorn der Familie oder der Nachbarn ein.



Der Eingang unserer Hardware wird an Port PB 7 des VIA 6522 angeschlossen, wie in Bild 4 ersichtlich. Mit diesem Interface läßt sich einiges an Verständnis für dieses doch recht komplexe VIA erarbeiten.

Geben Sie zuerst, nachdem das Interface angeschlossen wurde, folgendes Programm ein, und starten Sie es mit dem G-Command ab Adresse 0200:

Ø2ØØ	LDA	# 8Ø	DATENRICHTUNGSREGISTER BIT 7
Ø2Ø2	STA	AØØ2	ALS AUSGANG PROGRAMMIEREN
Ø2Ø5	LDA	# ØØ	AUSGANG PB 7
Ø2Ø7	STA	AØØØ	AUF Ø SETZEN
Ø2ØA	JSR	Ø219	VERZÖGERUNGSSUBROUTINE
Ø2ØD	LDA	# 3Ø	AUSGANG PB 7
Ø2ØF	STA	AØØØ	AUF 1 SETZEN
Ø212	JSR	Ø219	VERZÖGERUNGSSUBROUTINE
Ø215	JMP	Ø2Ø2	SPRUNG AN ANFANG

Ø218	NOP	
Ø219	LDX # FF	VERZÖGERUNGSRoutine:
Ø21B	DEX	INDEX X MIT FF LADEN
Ø21C	BNE Ø21B	X UM 1 ERNIEDRIGEN
Ø21E	RTS	WENN X ≠ Ø, DANN ERNEUTER SCHLEIFENDURCHLAUF
		RÜCKSPRUNG

Dieses Programm erzeugt einen Dauerton. Es besteht aus einer Schleife, die sich nur durch Drücken der Reset-Taste wieder verlassen läßt.

3.2. Ansteuerung des VIA 6522

Da unser Musik-Interface durch ein Rechteck-Signal an Portleitung PB 7 des VIA 6522 angesteuert wird, müssen wir uns etwas genauer ansehen, wie die softwaremäßige Ansteuerung dieses Bausteins vorgenommen wird. Das VIA wird von einem Programm angesteuert, wie ein RAM-Speicherbaustein, d. h. es können Daten im VIA abgespeichert oder vom VIA geladen werden.

Die Adressen des VIA liegen zwischen A000 und A00F, es sind also insgesamt 16 einzeln adressierbare Byte. Aus dem Anwenderhandbuch ist zu entnehmen, welche Zuweisung diese Adressen haben. Da das VIA hochkomplex und mit zahlreichen programmierbaren Funktionen ausgerüstet ist, genügt es uns vorerst zu verstehen, wie die 16 (2 mal 8) Ein-/Ausgangsleitungen von Port A und Port B anzusteuern sind. Da diese beiden Ports wahlweise als Eingang oder als Ausgang fungieren können, muß man zuerst festlegen, welche Pins Eingänge bzw. Ausgänge sind. Dies geschieht mit den beiden Datenrichtungsregistern (DDRB und DDRA). Datenrichtungsregister B ist unter Adresse A002 ansteuerbar, Datenrichtungsregister A unter Adresse A003. Ist in den Datenrichtungsregistern eine logische 0 abgespeichert, so ist das entsprechende Bit des Datenregisters ein Eingang, ist dieses Datenrichtungsregisterbit 1, so ist das dazugehörige Pin des Datenregisters ein Ausgang.

Sehen wir uns das Programmbeispiel zur Musikerzeugung an. Nach dem Einschalten oder nach Reset sind die beiden Datenrichtungsregister so programmiert, daß die

Bits der Datenregister Eingänge sind. Im Programm wird deshalb zuerst das Datenrichtungsregister B (DDRB), welches unter der Adresse A002 anzusteuern ist, mit 80 geladen. Dies macht den Port PB 7, den wir ja für die Musikerzeugung benötigen (und nur diesen!) zum Ausgang. Anschließend können wir das Port B Datenregister abwechselnd mit 80 (PB 7=1) und mit 00 laden. Die anderen Bit der Ports werden nicht verändert, denn sie sind ja als Eingänge programmiert. Durch das abwechselnde Umsteuern von PB 7 gibt diese Leitung ein Rechtecksignal an unsere Hardware ab.

Im nächsten Heft werden wir unser Musik-Interface mit einem zunehmend komplexeren Programm ansteuern. Bis dahin kann der Leser mit den nun gewonnen Erkenntnissen eigene Programmversuche unternehmen. Folgende Möglichkeiten zur Erstellung eines Programms bieten sich an:

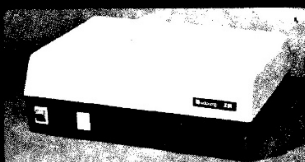
- 1.) Erzeugung eines sirenenartigen Tones
- 2.) Ansteuerung der Subroutine durch Tastenbedienung
- 3.) Erzeugen verschiedener Frequenzen durch Betätigen verschiedener Tasten
- 4.) Anzeige der jeweiligen Frequenz (Tonhöhe) auf dem Display

Fehlerberichtigung: In der letzten Folge hat sich ein Fehler eingeschlichen. Die Spannung an der Sekundärwicklung des Trafos für die 5 Volt-Versorgung sollte ca. 9 – 12 Volt betragen und nicht wie angegeben 3 – 12 Volt!

M. Krüger, Berlin ■

Preiswerte Normalpapierdrucker für 120mm- und DIN A4-Rollen

- X 50: 52 Stellen, Papier 120-mm-Rollen
- X 80: 80 Stellen, Papier 210-mm-Rollen
- Für PET 2001/CBM 3001 Voll adressierbar + Cursor- und Grafiksymbole
- Für SHARP MZ 80 K Alle Grafiksymbole. Interface enthalten
- Für RADIO SHACK TRS 80 Mit und ohne Extension-Box. Alle Symbole



PREISE: X 50 ab DM 1600,- (plus MwSt.)
X 80 ab DM 1900,- (plus MwSt.)

Hannover Messe '80
Stand 1900, Halle 18 OG.

- Für V24 RS232C, 20 mA Eingebauter Puffer, 50 bis 9600 baud
- ★ Neu: Für CBM lieferbar:
- Datenlogger für Forschung und Industrie
- IBM-kompatible Floppy, 8 Zoll, 1 Mio. Byte
- Winchester-Datenbank für 16 CBMs
- Interface für Präzisionswaagen

adcomp

Datensysteme GmbH · Horensstraße 8
8000 München 19 · Telefon 089/19 40 19

Lernen mit AIM, Teil 3

Eine "AIM-Einführung", Teil 3

Aufgaben in 2.4

In Abschnitt 2.4 wurden zwei Aufgaben gestellt, an Hand derer sich der noch 6502-unerfahrene Leser in der AIM-Programmierung üben konnte. Hier nun die Lösung der Aufgaben:

- a.) Ein Programm, das die Speicherzellen 0200 – 03FF mit einer fortlaufenden Zahl füllt:

0200 LDX # 00

0202 TXA

0204 LDA # 03

0206 STA 01

0208 LDY # 00

020A STY 00

020C TYA

020E STA (00),Y

0210 INY

0212 BNE 0209

0214 INC 01

0216 LDA # 10

0218 CMP 01

021A BNE 0208

021C JMP E182

Dieses Programm ist wohl das komplexeste von allen bisherigen Programmen, die wir besprochen haben. Das liegt daran, daß wir die indirekte Adressierung benutzen, denn diese Adressierungsart bietet sich in diesem Beispiel als günstig an.

0203 STA 0300,X

0206 INX

0207 BNE 0202

0209 JMP E182

Dieses Programm ist gegenüber dem in Abschnitt 2.1 vorgestellten sogar noch etwas kürzer, trotz ähnlicher Funktion.

- b) Das folgende Programm löscht die Speicherzellen von 0300 – 0FFF (sofern Ihr AIM mit 4K-RAM bestückt ist):

ANFANGSADRESSE (0300)

IN

PAGE SERO

00 UND 01

ACCU MIT 00 LADEN

SPEICHERBYTE (INDIREKT) LÖSCHEN

ZÄHLER ERHÖHEN

ENDEKRITERIUM ERREICHT?

SEROPAGEADRESSE 01 INCREMENTIEREN

UND LADEN

UND PRÜFEN, OB

ALLE BYTES GELÖSCHT

WENN JA, DANN ZURÜCK ZUM MONITOR

Wir wollen nun, zum Abschluß, eine selbstgeschriebene Subroutine besprechen, die alle Bytes in einem frei wählbaren Speicherbereich mit einem ebenfalls frei wählbaren Wert lädt. Wir speichern die benötigten Parameter in Page Zero ab:

ØØ	Anfangsadresse (low)
Ø1	Anfangsadresse (high)
Ø2	Endadresse (low)
Ø3	Endadresse (high)
Ø4	Zähler (low)
Ø5	Zähler (high)

Ø6 zu speichernder Wert

Wir benötigen diese Parameter unter anderem deswegen, weil die folgende Subroutine als universeller Software-Baustein verwendbar sein soll, den man in einem größeren Programm verwenden kann. Wenn Sie wollen, so können Sie sich unter Zuhilfenahme dieses Programmoduls und der Ihnen bereits bekannten I/O-Subroutinen an einem Speicherprüfprogramm versuchen.

Es ist allgemein üblich und eine gute Programmierweise, ein umfangreicheres Programm aus kleinen Modulen aufzubauen, die für sich alleine übersichtlich konzipiert und leicht auf Fehlerfreiheit getestet werden können. Aus mehreren dieser Programmbausteine lassen sich dann die komplexen Aufgaben leicht durch geschicktes Kombinieren lösen. Der 6502 ist ein Prozessor, der dem Programmierer diese Softwaretechnik förmlich aufdrängt, denn nur so kann man mit ihm absolut effektiv arbeiten, auch wenn die Subroutinen für sich alleine betrachtet vielleicht etwas länger werden.

Nun zu dem Programm:

```

Ø200 PHA      ACCU UND
Ø201 TXA      X-REGISTERINHALTE
Ø202 PHA      STACK
Ø203 LDA Ø0   ANFANGSADRESSE
Ø205 STA Ø4
Ø207 LDA Ø1   IN
Ø209 STA Ø5   ZÄHLER
Ø20B LDX #ØØ  X MIT ØØ LADE
Ø20D LDA Ø6   ACCU MIT WERT LADEN
Ø20F STA (Ø4,X) WERT IN AKTUELLER ZÄHLERADRESSE SPEICHERN
Ø211 LDA Ø4   ZÄHLER LOW UND ENDADRESSE
Ø213 CMP Ø2   LOW GLEICH?
Ø215 BEQ Ø227 WENN JA, SPRUNG
Ø217 CLC
Ø218 LDA Ø4   ZÄHLER
Ø21A ADC #Ø1
Ø21C STA Ø4   UM
Ø21E LDA Ø5
Ø220 ADC #ØØ  1
Ø222 STA Ø5   ERHÖHEN
Ø224 JMP Ø20D ZURÜCK, NÄCHSTE ADRESSE BEDIEHEN
Ø227 LDA Ø5   ZÄHLER HIGH UND
Ø229 CMP Ø3   ENDADRESSE HIGH GLEICH?
Ø22B BNE Ø217 WENN NICHT, DANN SPRUNG
Ø22D PLA      REGISTERINHALTE VOM
Ø22E TAX      STACK ZURÜCKHOLEN
Ø22F PLA
Ø230 RTS      (VORERST JMP E1821:)
```

3.3 Eine elektronische Sirene

Im letzten Heft wurde Ihnen vorgeschlagen, den AIM so zu programmieren, daß er über unseren Musikzusatz einen sirenenartigen Ton erzeugt. Wir wollen Ihnen heute ein solches Programm vorstellen, in dem ein Ton mit wechselnder Tonhöhe erzeugt wird:

```

Ø200 LDA #80   DATENRICHTUNGSREGISTERBIT PB7
Ø202 STA AØ02  ALS AUSGANG
Ø205 LDY #FF   INDEX Y AUF FF
Ø207 LDX #80   ROUTINE, DIE
Ø209 LDA #ØØ   255 mal (255 PERIODEN=IN-
Ø20B JSR Ø22D  HALT VOM Y-REGISTER)
Ø20E LDX #80   DURCHLAUFEN WIRD
Ø210 LDA #80   ERZEUGT TONHÖHE,
Ø212 JSR Ø22D  DIE DURCH WERT
Ø215 DEY       IN X-REGISTER
Ø216 BNE Ø207  VORGEGEBEN IST
Ø218 DEY       WIE
Ø219 LDX #B0   .
Ø21B LDA #ØØ   VORANGEHENDE
Ø21D JSR Ø22D  .
Ø220 LDX #B0   ROUTINE,
Ø222 LDA #80   .
Ø224 JSR Ø22D  NUR FÜR
Ø227 DEY       .
Ø22B BNE Ø219  ANDERE
Ø22A DEY       .
Ø22B BNE Ø207  TONHÖHE
Ø22D STA AØ0Ø  SUBROUTINE. STEUERT
Ø230 DEX       BIT 7 VON PORT B AN,
Ø231 BNE Ø23Ø  VERZÖGERUNGSSCHLEIFE, VORGEGEBEN
Ø233 RTS       DURCH INHALT VOM X-REGISTER (TONHÖHE)
```

3.4 Musikerzeugung mittels Taktgeber

Bisher hatten wir ein Rechtecksignal erzeugt, indem wir den logischen Zustand von Bit 7 des Port B im Rhythmus der erwünschten Frequenz wechselten. Mancher Leser fragte sich vielleicht, warum wir gerade PB7 als Ausgang zur Ansteuerung der Interface-Schaltung verwendet hatten. Dies geschah nicht ohne besonderen Grund. BP7 läßt sich nämlich ebenfalls durch einen der beiden Timer/Taktgeber des VIA 6522 ansteuern. Die nötige Software für die Timeransteuerung ist minimal, und es lassen sich Rechtecksignale mit hoher Frequenz und feiner Abstufung erzeugen. Ein weiterer Vorteil der Erzeugung von Impulsen mittels des Taktgebers ist der, daß die CPU nicht ständig mit sich wiederholenden Operationen beschäftigt ist, der Timer läuft, wenn er entsprechend programmiert ist, alleine weiter. So ließen sich etwa komplexere Signalformen erzeugen, oder, ohne allzu großen Aufwand, eben ein Musikprogramm mit Tastenbedienung, eine "Mini-Orgel".

Wie muß nun der Timer programmiert werden, um eine fortlaufende Folge von Impulsen zu erzeugen? PB7 kann nur durch Taktgeber 1 angesteuert werden. Im VIA 6522 befindet sich ein Register, das Hilfssteuerregister (engl.: Auxiliary Control Register = ACR), dessen Zustand neben anderen Funktionen die Arbeitsweise von Timer 1 steuert. Im ACR, welches softwaremäßig im AIM unter Adresse A00B zu erreichen ist, sind es die beiden höchstwertigen Bits, die für unseren Timer 1 zuständig sind.

Sind beide, Bit 6 und Bit 7, auf 1, so befindet sich der Timer in der freilaufenden Betriebsart, in welcher er mit dem 1MHz Taktsignal eine 16-Bit Binärzahl in den Registern T1L (niederwertiges Timer-Byte, im AIM unter Adresse A004) und T1H (höherwertiges Timer-Byte, Adresse A005) bis 0 herunterzählt, um dann den logischen Zustand von PB7 zu wechseln. Nun wird der Zyklus wiederholt. Studieren Sie zum Verständnis folgendes Programm:

```

0200 LDA #00      PB7 ALS AUSGANG
0202 STA A002      VORBEREITEN
0205 LDA #C0      HILFSSTEUERREGISTER AUF
0207 STA A00B      FREILAUFENDE BETRIEBSART U. PB7-ANSTEUERUNG
020A LDA #00      00 IN
020C STA A004      NIEDERWERTIGES TIMERBYTE (T1L)
020F LDA #02      02 IN HÖHERWERTIGES TIMERBYTE
0211 STA A005      (≈ 512 µs, d.h. ca. 976,6Hz)
0214 JMP E102      RÜCKSPRUNG ZUM MONITOR

```

3.5 Die Sirene mit fließendem Übergang

In 3.3 wurde Ihnen ein kurzes Programm vorgestellt, das eine Sirene mit zwei wechselnden Tönen erzeugte. Das nun folgende Beispiel läßt die Töne weich ineinander übergehen:

```

0200 LDA #00      PB7
0202 STA A002      UND
0205 LDA #C0      ACR
0207 STA A00B      VORBEREITEN
020A LDA #01      ZÄHLER FÜR TIMER 1
020C STA 0001      VORBEREITEN
020F STA A005      TIMER 1 STARTEN
0212 CLC
0213 LDA 00      VARIABLE
0215 ADC #01
0217 STA 00      FÜR

```

```

0219 TAX          ZÄHLER UM
021A LDA 01      1
021C ADC #00
021E STA 01      ERHÖHEN
0220 JSR 0246      SUBROUTINE ANSPRINGEN
0223 CPX #00      PRÜFEN, OB
0225 BNE 0212      ZÄHLERHÖCHSTWERT (0000)
0227 CMP #08
0229 BNE 0212      ERREICHT
022B SEC          JA, DANN
022C LDA 00      .
022E SBC #01      SUBTRAKTION
0230 STA 00      .
0232 TAX          VON VARIABLE
0233 LDA 01      .
0235 SBC #00      BIS
0237 STA 01      .
0239 JSR 0246      .
023C CPX #08      ZÄHLERTIEFSTWERT
023E BNE 022B      .
0240 CMP #05      .
0242 BNE 022B      ERREICHT
0244 BEQ 0212      -----
0246 PHA          SUBROUTINE,
0247 TXA          DIE
0248 PHA          DEN
0249 LDX #00      WERT
024B INX          DER
024C BNE 024B      VARIABLEN
024E LDA 00      IN
0250 STA A006      DIE
0253 LDA 01      ZÄHLERLATCHES
0255 STA A007      ÜBERGIBT
0258 PLA          UND
0259 TAX          ZEITVERZÜGERT
025A PLA
025B RTS          RÜCKSPRUNG

```

In den Adressen auf Page Zero 00 und 01 ist ein Zähler angelegt, der die jeweilige Frequenz des erzeugten Tones bestimmt (00=niederwertiges Byte, 01= höherwertiges). Dieser Zähler wird im Additionsprogrammteil ab Adresse 0212 erhöht, bis der erwünschte Endwert erreicht ist, um im anschließenden Subtraktions-

programmteil bis zum Anfangswert erniedrigt zu werden und so fort. Die Frequenz für Anfangs- und Endwert wird durch Vergleichsoperationen erzielt. Diese Vergleichsoperationen finden statt in:

Ø223 CPX #8Ø ENDWERT LOW

.

Ø227 CMP #Ø8 ENDWERT HIGH

Und

Ø23C CPX #Ø8 ANFANGSWERT LOW

.

Ø24Ø CMP #Ø5 ANFANGSWERT HIGH

Hier bieten sich viele Möglichkeiten zum Experimentieren an: Z. B. ließen sich die Vergleichsoperationen durch Ändern der Immediate-Adressierung in Zero-Page-Adressierung mit Werten aus dem RAM durchführen, etc.

Die Zeit, die zwischen Anfangs- und Endwert liegt, läßt sich durch Variieren des Wertes im Befehl

0249 LDX #80

in gewissen Grenzen bestimmen.

Beim Studium dieses Programms fällt auf, daß die neuen Werte für die Taktgeberzeitverzögerung nicht, wie vorher beschrieben, in die Register A004 (low) und A005 (high) des VIA geschrieben werden, sondern in A006 (low) und A007 (high).

Dies ist deswegen nötig, weil der Zähler mit dem Transfer eines Wertes nach Adresse A005 stets neu gestartet wird. Im Adresspaar A006 und A007 hingegen befinden sich zwei Latches, aus denen die Daten erst dann geladen werden, wenn der Zähler seinen Nulldurchgang hatte. So ergibt sich stets ein sauberer und korrekter Frequenzübergang. Der Zähler muß aber einmal zu Beginn gestartet werden! Dies geschieht in:

020F STA A005

16 ELCOMP

3.6 AIM - Mini - Orgel

Wie bereits im letzten Heft versprochen, werden wir unseren Lerncomputer zu einem "Musikinstrument" umfunktionieren. Diese Aufgabe dürfte uns nach allem bisher besprochenen keine allzu großen Schwierigkeiten bereiten. So wird denn auch das dazu nötige Programm relativ kurz. Allerdings arbeitet dieses Programm mit zwei Tabellen, wobei die erste dazu dient, dem ASCII-Tastenwert, mit dem der ACCU von der Monitor-Subroutine RCHEK zurückkehrt, einen Hex-Wert im Indexregister X zuzuordnen.

Dieser Wert im X-Register wird dann (siehe folgendes Programm ab 021C) um 1 erhöht (er ist dann ≥ 1), und verdoppelt (durch ASL A). Nun kann das X-Register als Zeiger dienen, um aus der zweiten Tabelle den nötigen 2-Byte Wert für die Taktgeberlatches zu holen und in diesen abzuspeichern. Im folgenden nun zuerst die Tabellen und dann das Programm:

Ø3ØØ ØØ 4A 48 47

Ø3Ø4 46 44 53 41

Ø3Ø8 55 59 54 52

Ø3ØC 45 57 51 37

Ø31Ø 36 35 34 33

Ø314 32 31

Ø316 ØØ ØØ Ø5 Ø2

Ø31A 38 Ø2 7E Ø2

Ø31E FØ Ø2 F6 Ø2

Ø322 53 Ø3 BC Ø3

Ø326 F4 Ø3 7Ø Ø4

Ø32A FC Ø4 98 Ø5 TABELLE 2

Ø32E ED Ø5 A6 Ø6

Ø332 78 Ø7 E9 Ø7

Ø336 E1 Ø8 F7 Ø9

Ø33A 32 ØB DA ØB

Ø33E 4E ØD EF ØE

TABELLE 1

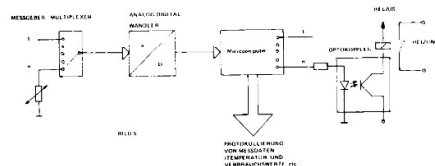
TABELLE 2

Lernen mit AIM, Teil 4

Eine "AIM-Einführung", Teil 4

4.0 Analog- und Digitalsignale

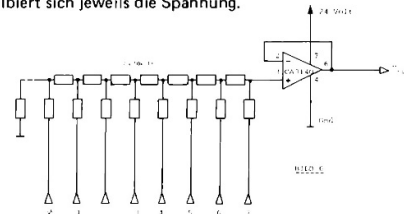
Nachdem wir in der letzten Folge Hardware und Software zur Erzeugung musikhähnlicher Klänge kennengelernt haben, wollen wir uns nun dem Problem der Umwandlung analoger Signale in digitale und umgekehrt zuwenden. Mit dem ständigen Fortschreiten der Elektronik gewinnt dieser Bereich der Digitalelektronik besonders in Anwendung als Microcomputerperipherie an zunehmender Bedeutung. Die Informationen, die in unserer Umwelt anfallen, sei diese Umwelt natürlich oder technisch, sind zumeist analoger Natur. Mit dem zunehmenden Einsatz von Microcomputern für Aufgaben des Messens, Steuern und Regels ist es immer mehr nötig, diese analogen Informationen in für unseren Rechner begreifbare digitale Signale umzuwandeln. Ebenso müssen die Digitalzahlen, mit denen der Computer arbeitet, in vielen technischen Bereichen in analoge Signale zurückverwandelt werden. Beispiele, in denen der Rechner die Fähigkeit zur A/D und D/A-Wandlung benötigt, sind: Spracherkennung, Signalerzeugung mit beliebiger Kurvenform, erfassen von Meßdaten, Heizungs- und Motorsteuerung und vieles andere mehr. Bild 5 zeigt das Blockdiagramm einer Anlage zur Erfassung von Temperaturwerten in einem Wohnhaus. Über Schaltausgänge lassen sich die Heizelemente des Hauses betätigen. Diesem Beispiel kommt für die Anwendung eines Hobbycomputers besondere Bedeutung zu. Ist es doch hiermit möglich, bestimmte Temperaturgrenzwerte vorher einzugeben, die je nach Raum und Tageszeit differieren, und so den Energieverbrauch des Hauses minimieren.



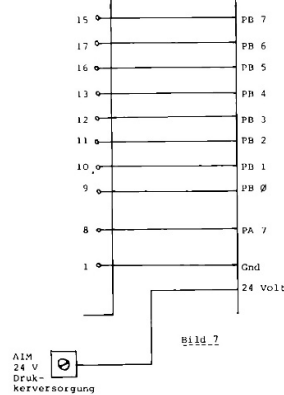
4.1 Das Prinzip der Digital-Analogwandlung

Unter den vielen technischen Prinzipien zur Umwandlung digitaler Signale und Analoge wurde eines ausgewählt, welches häufig verwandt wird, sich mit wenigen preiswerten Bauteilen aufbauen läßt, hervorragend an den VIA 6522 anzuschließen ist, und als Modell einen äußerst anschaulichen Charakter hat. Bei der Auslegung der Schaltung wurde davon ausgegangen, daß in der AIM-Grundversion nur zwei Versorgungsspannungen zur Verfügung stehen, 5 Volt und 24 Volt, und daß der Wandler sich mit der 24 Volt Versorgungsspannung begnügt.

Bauen Sie die Schaltung nach Bild 6 auf und schließen Sie diese an den AIM nach Zeichnung 7 an. Wie funktioniert dieser D/A-Wandler? Im Prinzip besteht er nur aus einer Widerstandskette. Es empfiehlt sich, hier Widerstände mit höherer Genauigkeit, z. B. 1% Toleranz, zu verwenden, um die widerstandsbedingte Ungenauigkeit gering zu halten. An den Knotenpunkten dieser Widerstandskette werden die Digitalsignale, welche vom VIA 6522 kommen (diese sind von relativ konstanter Spannung) eingespeist. Das höchstwertige Bit an Portleitung PB 7 hat dabei den größten Einfluß, in Richtung zu PB 0 halbiert sich jeweils die Spannung.



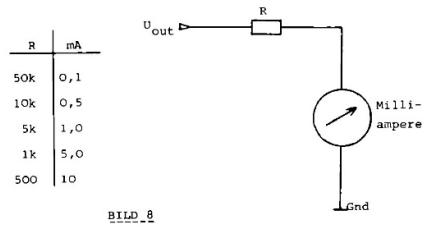
J1 - Application D/A - Interface



Der Operationsverstärker CA 3140 ist hier als Spannungsfolger beschaltet, dies bedeutet, daß die am Ende der Widerstandskaskade anliegende Spannung weitgehend unabhängig ist von der Belastung am Ausgang unserer Schaltung. Der Operationsverstärker wird durch die 24 Volt Spannung des AIM-Druckers versorgt. Die Genauigkeit des Wandlers hängt außer von der Widerstandstoleranz noch von der Stabilität der 5 Volt-Versorgung ab. Daß der binäre Wert 0 nicht genau auf 0 Volt liegt, ist prinzipbedingt. Doch die wenigen Millivolt, die der Realwert bei Verwendung des Präzisionsverstärkers CA 3140 vom Nullpunkt abweicht, können bei dieser einfachen Schaltung wohl schon einmal in Kauf genommen werden. Uns kommt es ja in erster Linie auf den Lerneffekt an – und dennoch ist dieser Wandler für viele Anwendungsfälle ausreichend genau genug.

4.2 Software für den D/A-Wandler

Haben Sie die Schaltung richtig aufgebaut, so wird die softwaremäßige Ansteuerung kein Problem sein. Schon mit den Monitorroutinen können wir jetzt analoge Signale erzeugen. Zuerst müssen wir mit dem M-Command unter Adresse A002 die Datenrichtung der Port B Leitungen als Ausgänge festlegen (mit FF, wie üblich). Um die Analogspannung am Ausgang des D/A-Wandlers zu erzeugen, kann nun ein Zahlenwert in Adresse A000 eingeschrieben werden. Zur Überprüfung des Spannungspiegels empfiehlt es sich, ein Vielfachmeßinstrument an den Analogausgang der Schaltung anzuschließen. Auch auf dem Oszilloskop kann man die Analogsignale, die man erzeugt, anschaulich darstellen. Besitzen Sie weder Oszilloskop noch Multimeter, so können Sie auch ein kleines Zeigerinstrument kaufen und an den Analogausgang entsprechend Bild 8 anschließen.



Für ein Milliampere-Meter muß dann noch ein Vorschaltwiderstand in den Stromkreis eingefügt werden. Den rechten Wert für diesen Widerstand entnehmen Sie der Tabelle in Bild 8.

Um einfache Kurvenformen zu erzeugen, müssen wir nun ein Programm schreiben. Geben Sie beispielsweise ein:

```

Ø2ØØ LDX #FF
Ø2Ø2 STX AØØ2
Ø2Ø5 INX
Ø2Ø6 STX AØØØ
Ø2Ø9 JMP Ø2Ø5

```

Ohne Oszilloskop, nur mit dem Zeigerinstrument oder dem Voltmeter sehen Sie wahrscheinlich garnichts, bestenfalls ein Zittern des Zeigers oder ein Flattern der Anzeige (bei Digitalinstrumenten), der Rechner ist einfach zu schnell.

Ohne große Mühe läßt sich aber vor dem Sprungbefehl ein Subroutinesprung zu einer Verzögerungsroutine einbauen, die das Programm so stark verlangsamt, daß man auch auf einem Zeigerinstrument noch sehen kann, wie sich das Programm auf den Analogausgang auswirkt.

Dieses Programm sieht dann folgendermaßen aus:

```

Ø2ØØ LDX #FF
Ø2Ø2 STX AØØ2
Ø2Ø5 INX
Ø2Ø6 STX AØØØ
Ø2Ø9 JSR Ø2ØF

-----
Ø2ØC JMP Ø2Ø5
Ø2ØF TXA
Ø21Ø PHA
Ø211 LDX #ØØ
Ø213 LDY #ØØ
Ø215 INX
Ø216 BNE Ø215
Ø218 INY
Ø219 BNE Ø215
Ø21B PLA
Ø21C TAX
Ø21D RTS

```

Zum Abschluß des Kapitels D/A-Wandlung noch ein Programm zur Sinuserzeugung:

```

Ø2ØØ LDX #FF
Ø2Ø2 STX AØØ2
Ø2Ø5 INX
Ø2Ø6 LDA Ø3ØØ,X
Ø2Ø9 STA AØØØ
Ø2ØC JMP Ø2Ø5

```

Dieses Programm benutzt eine Tabelle, die unter Adresse 0300–03FF angelegt sein muß. In dieser Tabelle sind die Zahlenwerte der Sinusfunktion abgespeichert. Nachfolgend finden Sie einen Druckerauszug, der mit dem AIM-Drucker erstellt wurde und die einzelnen Kurvenpunkte in ihrer Speicherplatzzuweisung enthält. Natürlich wurden die einzelnen Daten nicht mit dem Taschenrechner ausgerechnet, um sie dann von Hand einzugeben, dazu wäre ich viel zu faul!

Mit dem AIM-BASIC hat die Ermittlung der einzelnen Werte jedoch keine große Mühe gemacht.

```

<M>=0300 79 78 7E 81
<> 0304 84 87 8A 8D
<> 0308 90 93 96 99
<> 030C 9B 9E A1 A4
<> 0310 A6 A9 AC AE
<> 0314 B1 B4 B6 B9
<> 0318 BB BE C0 C2
<> 031C C5 C7 C9 CB
<> 0320 CD CF D1 D3
<> 0324 D5 D7 D9 DB
<> 0328 DE DF E1
<> 032C E2 E4 E5 E6
<> 0330 E7 E8 E9 EA
<> 0334 EB EC ED EE
<> 0338 EE EF EF F0
<> 033C F0 F0 F0 F0
<> 0340 F0 F0 F0 F0
<> 0344 F0 F0 EF EF
<> 0348 EE EE ED EC
<> 034C EB EA E9 E8
<> 0350 E7 E6 E5 E4
<> 0354 E2 E1 DF DE
<> 0358 DC DB DA D7
<> 035C DB DA D1 D7
<> 0360 DB DB DB D7
<> 0364 C5 C2 C0 BE
<> 0368 BB B9 B6 B4
<> 036C B1 AE AC A9
<> 0370 A6 A4 A1 9E
<> 0374 9B 99 96 93
<> 0378 90 8D 8A 87
<> 037C 84 81 7E 7B
<> 0380 79 76 73 70
<> 0384 6D 6A 67 64
<> 0388 61 5E 5B 58
<> 038C 56 53 50 4D
<> 0390 4B 48 45 43
<> 0394 40 3D 3B 38
<> 0398 36 33 31 2F
<> 039C 2C 2A 28 26
<> 03A0 24 22 20 1E
<> 03A4 1C 1A 18 16
<> 03A8 15 13 12 10
<> 03AC 0F 0D 0C 0B
<> 03B0 0A 09 08 07
<> 03B4 06 05 04 03
<> 03B8 03 02 02 01
<> 03BC 01 01 01 01
<> 03C0 01 01 01 01
<> 03C4 01 01 02 02
<> 03C8 03 03 04 05
<> 03CC 06 07 08 09
<> 03D0 0A 0B 0C 0D
<> 03D4 0F 10 12 13
<> 03D8 15 16 18 1A
<> 03DC 1C 1E 20 22
<> 03E0 24 26 28 2A

```

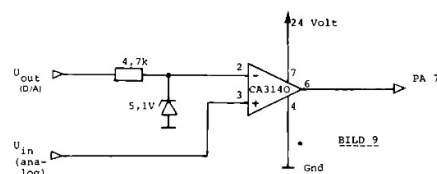
```

<> 03E4 2C 2F 31 33
<> 03E8 36 38 3B 3D
<> 03EC 40 43 45 48
<> 03F0 4B 4D 50 53
<> 03F4 56 58 5B 5E
<> 03F8 61 64 67 6A
<> 03FC 6D 70 73 76

```

4.3 Erfassen der analogen Umwelt: Der A/D-Wandler

Einen Wandler für analoge Eingangssignale bekommen wir, wenn wir unseren D/A-Wandler noch durch einen Komparator erweitern. Dieser Komparator, den wir ebenfalls mit dem Operationsverstärker CA 3140 realisieren können, hat die Aufgabe, die analoge Eingangsspannung, deren Wert wir ermitteln wollen, mit einer zweiten Spannung zu vergleichen, die wir im D/A-Wandler selbst erzeugen. Ist der Wert der im D/A-Wandler erzeugten Spannung größer als der Meßwert, so ist der Ausgang des Komparators High, im anderen Fall ist er Low. Andere Ausgangspegel außer "H" und "L" kann so ein Komparator praktisch nicht erzeugen. Die Schaltung des Komparators entnehmen Sie bitte Bild 9.



Widerstand und Zenerdiode in der Eingangsleitung dienen lediglich zum Überspannungsschutz, sollten aber bei einem Operationsverstärker wie dem 3140 mit extrem hochohmigem Eingang in jedem Fall vorhanden sein. Den Ausgang der Schaltung legen wir auf PA 7 des VIA 6522.

4.4 Analog-Digitalumwandlung: Das Zählprinzip

Mit Software realisieren wir nun das, was früher in den A/D-Wandlern aufwendige Logik- und Zählerschaltungen zu lösen hatten. Ein Prinzip zur Ansteuerung des Wandlers ist das Zählprinzip. Das geht so: Wir fangen bei 00 an, speichern diesen Wert in den acht Bit des Port B ab und überprüfen nun den Logikpegel an PIN PA 7. Ist dieser 1, so beträgt unsere Eingangsspannung 0 Volt oder weniger. Ist der Wert 0, so müssen wir den Zählwert um 1 erhöhen. Wir prüfen erneut den Wert in PA 7. Das geht so weiter, bis FF (255 dezimal) erreicht ist. Ist nun PA 7 immer noch low, so ist die zu ermittelnde Eingangsspannung größer als die maximale mit dem D/A-Wandler erreichbare Spannung (die etwa bei 4,8 – 5 Volt liegen müßte, muß experimentell ermittelt werden!) Wir müssen nun den Wandelvorgang abbrechen. Den Spannungswert, der proportional zum ermittelten Zählwert ist, erhalten wir, indem wir die maximal mit dem D/A-Wandler erreichbare Spannung durch 255 teilen und mit dem Zählwert multiplizieren. Das hört sich im Moment kompliziert an, ist aber in der Praxis recht einfach. Sehen Sie sich dazu die folgende Subroutine an:

Ø3ØØ	LDX	#FF	DDRB AUF FF
Ø3Ø2	STX	AØØ2	(PORT B - LEITUNGEN WERDEN AUSGAENGE)
Ø3Ø5	INX		X - REGISTER WIRD ØØ
Ø3Ø6	STX	AØØØ	X - WERT IN PORT B
Ø3Ø9	BIT	AØØ1	PA 7 TESTEN
Ø3ØC	BMI	Ø312	WENN 1, DANN ENDE (SPRUNG)
Ø3ØE	CPX	#FF	X TESTEN, WENN MAXIMALWERT NICHT
Ø31Ø	BNE	Ø3Ø5	ERREICHT, ERNEUTER SCHLEIFENDURCHLAUF
Ø312	RTS		RUECKKEHR INS HAUPTPROGRAMM

Sie sehen, die benötigte Software ist gering. Die Subroutine wird mit dem Wert im X-Register verlassen. Hierin steht er dem Hauptprogramm, welches diese Subroutine benutzt, zur Verfügung.

Im nächsten Heft werden wir dieses Thema fortsetzen. Es soll ein zweites Verfahren zur softwaremäßigen Ansteuerung des A/D-Wandlers vorgestellt werden, das zwar schneller als das Zählverfahren ist, jedoch ein etwas längeres Programm benötigt und komplizierter zu programmieren ist. Außerdem soll ein Hardwarevorschlag

gemacht werden, mit dem die Messung von Temperaturen möglich ist, sowie die Möglichkeit der Ansteuerung von Relais über Optokoppler (z.B. für die Heizungssteuerung) gezeigt werden.

Meine Programmieraufgabe zum nächsten Heft:
Entwickeln Sie ein Programm, das den AIM die Funktion eines einfachen Digitalvoltmeters realisieren läßt. Bis dann.

Berichtigung zu ELCOMP Nr. 5, Seite 18.

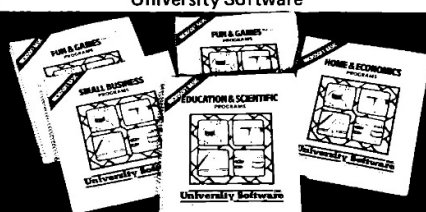
M. Krüger ■

Bei Adresse 020D muß es heißen:

LDA #80 (anstelle 30)

Dieses Programm ist eine Subroutine. Es muß daher mit einem Subroutinesprung angesprungen werden.

In letzter Minute eingetroffen!!!
Brandneu aus USA!!
University Software



Application Programs — published in Microsoft BASIC (für Sorcerer Computer) kann auf andere Computer mit Microsoft BASIC leicht umgeschrieben werden.
5 Bände mit Spiralbindung
Small Business Programs DM 199,-
Education + Scientific DM 139,-
Fun + Games, Vol. I DM 59,-
Fun + Games, Vol. II DM 59,-
Home + Economics Programs DM 99,-
Die gesamte Bibliothek kann für DM 543,- bezogen werden. Wir sind Exklusivvertreter für Zentraleuropa.
Lieferung ab Ende April. Vorbestellungen schon jetzt möglich.
Lieferung per NN oder Vorkasse. Postscheckkonto München Nr. 15 994-807.
Ing. W. Hofacker GmbH Verlag, Tegernseerstr. 18
D-8150 Holzkirchen, Tel.: 08024/7331

PIC PROGRAMMODULE FÜR INDIVIDUELLE COMPUTERSOFTWARE
PIC-RECHENZENTRUM
PARKSTR. 30
4174 SEVELEN

PIC-GmbH, Europas erstes Softwareversandhaus
Standardprogramme für
CBM und TRS 80 ab 490,— DM

Lager-, Adress-, Karteiverwaltung, Buchführung, Lohnabrechnung, Baulohn.
Informationsmappe gegen 50 DM Schutzgebühr
Handbücher 100 DM (wird bei Programmkau angeordnet)
Sehr viel Zubehör ab Lager
(u.a. Basic-Editor, Lautsprecher, Notstromversorgung)
Preislisten gegen fertig adressierten A 5 Umschlag und 5 DM (in Briefmarken oder bar)

Lernen mit AIM, Teil 5

Lernen mit dem AIM Eine "AIM-Einführung", Teil 5

Aufgabe aus 4.4

Im letzten Heft wurde Ihnen der Vorschlag gemacht, ein Programm zu entwickeln, das das Gerät zu einem simplen Digitalvoltmeter umfunktioniert. Dieses Programm konnte unter Zuhilfenahme des A/D-Wandlers und der Subroutine in 4.4, die den A/D-Wandler nach dem als Zählverfahren bekannten Prinzip ansteuert, konzipiert werden. Als diese Aufgabe gestellt wurde, glaubte man, daß ein aufwendiges Programm dazu notwendig sei. Aber beim Programmieren wurde festgestellt: Die Software ist ganz einfach! Sehen Sie selbst:

```
0200 LDX #00
0202 STX 00      SEROPAGEBYTES 00 UND 01
0204 STX 01      LÖSCHEN
0206 CLC        CARRYBIT LÖSCHEN
0207 SED        AUF DEZIMALADDITION VORBEREITEN
0208 JSR 0300    SUBROUTINE AUS 4.4. ANSPRINGEN
020B CPX #00     WENN X-REGISTER = 0,
020D BEQ 021E    DANN ENDE
020F LDA 00      ZÄHLER
0211 ADC #19     .
0213 STA 00      DEZIMAL
0215 LDA 01      .
0217 ADC #00     UM 19 ERHÖHEN
0219 STA 01      ERHÖHEN
021B DEX        X-REGISTER DECREMENTIEREN
021C BNE 020F    WENN ≠ 0, DANN NOCH EINMAL
021E CLD        DEZIMALMODE LÖSCHEN
021F JSR EA13    CRL0W+
0222 JSR E83B    BLANK2+
0225 LDA 01      ZÄHLER, HÖHERWERTIGES BYTE
0227 JSR EA46    NUMA+
022A LDA 00      ZÄHLER, NIEDERWERTIGES BYTE
022C JSR EA46    NUMA+
022F JMP E182    ZURÜCK ZUM MONITOR
```

+: Siehe AIM-Anwenderhandbuch

Das Programm benutzt einen Trick: Das Ergebnis der Analog-Digitalumwandlung wird durch einen Hexadezimalwert im X-Register aus dem Unterprogramm zurück ins Hauptprogramm gebracht. Die Zahl im X-Register 00 bis FF (= 0 – 255 dez.) entspricht einem Spannungswert am Analogeingang zwischen 0 Volt und 4,9 Volt. Jede

Stufe des X-Registers entspricht daher:

4,9V = 4900 Millivolt

4900 : 255 = ca. 19

19 Millivolt. Das Programm geht daher so vor, daß der Wert im X-Register um Eins vermindert wird, dafür wird ein Zähler um 19 dezimal erhöht. Sehen Sie sich dazu das Programm an! Die Speicherstellen für den Dezimalzähler sind in der Zeropage auf 00 (low) und 01 (high).

Diese Programmierweise hat den Vorteil, daß das Programm relativ kurz und überschaubar ist, jedoch wird dieser Vorteil mit einer relativ langen Ausführungszeit (Schleifendurchlauf max. 255-fach) erkauft. Die Ausführungszeit spielt in unserem Fall eine untergeordnete Rolle.

Die Anzeige des Spannungswertes erfolgt in Millivolt. Der Rest des Programms nutzt wieder einmal die effektiven Monitorroutinen des AIMS zur Datenausgabe.

Nun noch eine Bemerkung zur Genauigkeit unseres Digitalvoltmeters. Natürlich kann man nicht die Präzision eines handelsüblichen Instrumentes erwarten, dazu müßte ein erhöhter Aufwand betrieben werden. Dennoch läßt sich die Funktionsweise des digitalen Meßprinzips an unserem Modell ein wenig verständlich machen. Die Genauigkeit des Meßwertes ist der eines einfachen analogen Voltmeters durchaus gleichwertig.

4.5 Sukzessive Approximation

Neben dem Zählverfahren ist in der digitalen Meßtechnik ein anderes Prinzip gebräuchlich, um schnelle A/D-Wandlungen durchzuführen. Dieses Verfahren wird in unserem Rechner nicht unbedingt nötig sein, denn für schnelle Meßwertmessungen wäre sowieso eine analoge Halteschaltung am Analogeingang nötig. Doch der Vollständigkeit halber soll die Sukzessive Approximation hier erwähnt werden. Sukzessive Approximation bedeutet sinngemäß soviel wie schrittweise Annäherung. Es wird ebenso wie beim Zählverfahren mittels eines D/A-Wandlers eine Referenzspannung erzeugt, die mit der zu ermittelnden Analogspannung verglichen wird. Stimmen beide Spannungen überein, so entspricht das Digitalwort, das in den D/A-Wandler eingegeben wurde, dem Wert der Analogeingangsspannung. Bei der Sukzessiven Approximation wird nun nicht eine Spannungsrampe durch Zählung erzeugt, sondern die einzelnen Bits des Digitalwortes werden hintereinander geprüft. Bei einem 8-Bit-Wandler erfolgt die Wandlung in 8 Stufen, (beim Zählverfahren bräuchte man bis zu 256 Stufen). Man fängt beim höchstwertigen Bit an. Dieses wird gesetzt und durch D/A-Wandlung geprüft, ob der sich ergebende Spannungswert größer ist als die Eingangsspannung. Ist er größer, so wird das geprüfte Bit wieder gelöscht, ist

der Spannungswert kleiner, so bleibt das Bit stehen. Nun wird, ganz genau wie eben beschrieben, das nächstniederwertige Bit geprüft und so fort, bis zum niedrigsten Bit. Das folgende Programm, welches wieder als Subroutinenmodul aufgebaut ist, arbeitet nach diesem Verfahren:

```

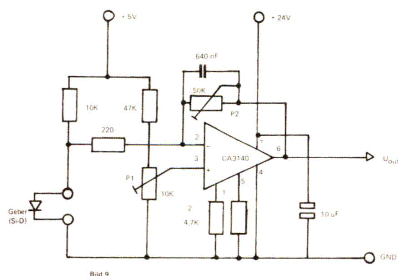
0320 LDX #FF      DDRB
0322 STX A002     VORBEREITEN
0325 INX
0326 STX 02       SEROPAGEBYTE 02 LÖSCHEN
0328 TXA
0329 SEC          CARRYBIT SETZEN ("SCHIEBEIT")
032A ROR 02       SEROPAGEBYTE 02 ALS SCHIEBEREGISTER BENUTZEN
032C TAX          IM X-REGISTER ZWISCHENSPEICHERN
032D ORA 02       SCHIEBEREGISTER U. ACCU VERKNÜPFEN
032F STA A000     ACCUINHALT IN D/A-WANDLER
0332 LDY #F0      ZEITVERZÖGERUNGSSCHLEIFE
0334 INY          ZUM EINSCHWINGEN
0335 BNE 0334     DES WANDLERS
0337 BIT A001     KOMPARATOR TESTEN
033A BPL 033D     WENN ANALOGWERT NICHT ERREICHT - SPRUNG
033C TXA          ANALOGWERT ERREICHT: ALTEN WERT VOM X LADEN
033D BCC 032A     WENN SCHIEBEREGISTER DURCHLAUFEN, DANN ENDE
033F TAX          WERT IN X, FÜR HAUPTPROGRAMM
0340 RTS          RÜCKSPRUNG

```

Sie können diese Subroutine anstelle des Moduls aus 4.4 in Verbindung mit dem Programm "Digitalvoltmeter", das zuvor beschrieben wurde, verwenden. Nicht vergessen: Sprungadresse ändern!

4.6 Temperaturmessung

Im letzten Heft wurde das Beispiel der Heizungssteuerung eines Wohnhauses mittels Microcomputer für einen sinnvollen Anwendungsfall erwähnt. Wollen wir eine solche Anlage simulieren, so muß der Rechner in der Lage sein, die Temperaturen der Umgebung mit dem A/D-Wandler in digitale Zahlenwerte umzuformen. Bild 9 zeigt das Schaltbild eines einfachen Meßverstärkers, den wir benötigen, um ein Eingangssignal von einigen Volt für den Wandler zu erhalten. Als Thermosensor dient in der Schaltung eine einfache Siliziumdiode. Diese ändert ihre Durchlaßspannung pro Grad Celsius um etwa 2,5 mV. Der Operationsverstärker vergleicht diese Spannung mit einer Referenzspannung, die vom Trimpotentiometer P1 gebildet wird, und verstärkt die entsprechende Differenz.



Der Aufbau dieser Schaltung ist nicht ohne Probleme, da sie wegen der hohen Verstärkung zum Schwingen neigt. Der Leser, der noch wenig Erfahrung besitzt im Aufbau analoger Schaltungen, sei darauf verwiesen, sich im Handel einen Bausatz für wenige Mark zu beschaffen, der für diese Zwecke gut geeignet ist.

Ein wenig diffizil gestaltet sich der Abgleich der Schaltung. Mit P1 stellen wir die Schaltung auf eine niedrige Temperatur (z. B. Zimmertemperatur) ein. Dann erwärmen wir den Meßfühler (etwa unter der Schreibtischlampe) und gleichen mit P2 ab. Zum Vergleich benutzen wir jeweils ein gutes Zimmerthermometer. Mehrmaliges abwechselndes Einstellen von P1 und P2 ist nötig, bis der erwünschte Meßbereich "sitzt", so wie Sie ihn für Ihren Anwendungsfall haben wollen. Als Hilfsmittel bietet sich entweder ein handelsübliches Vielfachmeßinstrument oder das Digitalvoltmeterprogramm aus diesem Artikel an.

Software zu diesem Hardwarezusatz ist nicht verfügbar, zu unterschiedlich sind die individuellen Einsatzmöglichkeiten. Das entsprechende Programm können Sie leicht selber schreiben, etwa durch Modifikation des Digitalvoltmeterprogramms oder nach Ihren eigenen Vorstellungen. Nun viel Spaß!

M. Krüger ■

In letzter Minute eingetroffen!!!
Brandneu aus USA!!
University Software

Application Programs — published in Microsoft BASIC (für Sorcerer Computer) kann auf andere Computer mit Microsoft BASIC leicht umgeschrieben werden.

5 Bände mit Spiralbindung

Small Business Programs	DM 199,-
Education + Scientific	DM 139,-
Fun + Games, Vol. I	DM 59,-
Fun + Games, Vol. II	DM 59,-
Home + Economics Programs	DM 99,-

Die gesamte Bibliothek kann für DM 543,- bezogen werden. Wir sind Exklusivvertreter für Zentraleuropa.

Lieferung ab Ende April. Vorbestellungen schon jetzt möglich.

Lieferung per NN oder Vorkasse. Postscheckkonto München Nr. 15 994-807.

Ing. W. Hofacker GmbH Verlag, Tegernseerstr. 18
D-8150 Holzkirchen, Tel.: 08024/7331

Lernen mit AIM, Teil 6

Lernen mit dem AIM, Teil 6

5.0 Die AIM-Uhr

Viele Heimcomputer, wie beispielsweise der CBM, besitzen eine interne Uhr. Hierbei kann der Bediener aus dem BASIC heraus eine Zeiteingabe vornehmen. Die einmal eingegebene Zeit wird kontinuierlich weitergezählt, ohne daß dabei die Programmierung oder Ausführung eines Programmes in irgendeiner Weise gestört wird. Auf Bedarf steht die aktuelle Uhrzeit dem Anwender des Rechners zur Verfügung.

Natürlich bietet auch unser AIM alle Voraussetzungen für die Erstellung einer solchen internen Uhr, denn dies ist für einen Mikroprozessor kein großes Problem, eine Aufgabe, die er "nebenbei" löst, so daß schon zahlreiche mikroprozessorgesteuerte Konsumgeräte, wie Rundfunk- und Fernsehempfänger, Küchenherd und Taschenrechner, mit diesem Service aufwarten können.

5.1 Der 6522 im Interruptbetrieb

Die Uhrenfunktion soll, so wäre es ideal, neben dem eigentlichen Rechnerbetrieb ablaufen. Egal, ob BASIC, Assembler, Editor, Monitor oder Anwenderprogramm, die Uhrzeit müßte in einem Speicherplatz ständig mit aktuellem Werte zur Verfügung stehen, so daß jedes Programm nach Bedarf darauf zurückgreifen kann. Wünschenswert wäre auch, wenn auf Drücken einer bestimmten Taste, etwa der F1-Taste, die genaue Uhrzeit angezeigt würde.

Diese Forderung ist natürlich nur mit Hardware außerhalb der eigentlichen Zentraleinheit zu realisieren.

Entweder müßte eine vom Rechner völlig unabhängige Uhrenschaltung vorhanden sein, oder ein Taktgeber erzeugt ein Signal, daß den Prozessor in zeitgleichen Intervallen unterbricht, um die Uhr, die etwa durch einige Speicherplätze im RAM realisiert werden könnte, einen Schritt weiterzuzählen. Dieser zweite Weg bietet sich beim AIM an, denn die quartzgenaue Steuerung des VIA 6522, den wir bereits in anderem Zusammenhang kennengelernt haben, ermöglicht es uns, einen Hardwareinterrupt in festgelegten Intervallen zu erzeugen.

Sehen wir uns an, wie der VIA 6522 zu programmieren ist, um ihn in die Interruptbetriebsart zu setzen:

0F30	LDA	#40	40 (BIT 6) IN
0F32	STA	A00B	ACR (AUXILIARY CONTROL REGISTER)
0F35	LDA	#C0	C0 (BIT 5 U. 7) IN
0F37	STA	A00E	IER (INTERRUPT ENABLE REGISTER)
0F3A	LDA	#1E	TIMER 1 - LATCHES
0F3C	STA	A004	MIT DEM
0F3F	LDA	#4E	ENTSPRECHENDEN WERT
0F41	STA	A005	LADEN
0F44	LDA	#00	STARTADRESSE DES
0F46	STA	A400	INTERRUPTORGRAMMS
0F49	LDA	#0F	IST
0F4B	STA	A401	0F00 (IRQ4)
0F4E	CLI		INTERRUPTFLAG IM STATUS LOESCHEN
0F4F	JMP	E182	ZUM MONITOR

Das ist alles, um den Timer auf Interruptbetrieb zu setzen, so daß alle 20 ms eine Unterbrechung erzeugt wird (sie dürfen dieses Programm aber erst starten, wenn ab Adresse 0F00 das Uhrenprogramm steht, sonst verschwindet die CPU auf ungewollten Bahnen!)

In diesem Initialisierungsprogramm wird zuerst das Auxiliary Control Register für freilaufende Betriebsart von Timer 1 programmiert. Anschließend wird das Interrupt Enable Register auf Interrupt von Timer 1 geschärft. Dann werden die Timer 1-Latches mit dem gewünschten Wert geladen, in unserem Fall für 20 ms = 50 Interrupts/sec. Im vierten Schritt wird die Adresse, welche nach einem Interrupt angesprungen wird (nachdem ein Stück im Monitor durchlaufen wurde), vorgegeben, in unserem Fall 0F00. Dies geschieht durch Abspeichern der Startadresse der Interruptroutine im IRQV4. IRQV4 ist eine Adresse A400/A401.

Letztlich bleibt nur noch übrig, das Interruptflag im Statusregister auf 0 zu setzen und zum Monitor zurückzukehren.

5.2 Die Zählerroutine

Im Prinzip ist der Kern unseres Uhrenprogrammes nichts weiter als ein kleines Zählerprogramm, das durch den Interruptimpuls vom

VIA initialisiert wird, je nach Bedarf eine oder mehrere Speicherzellen hochzählt bzw. zurücksetzt, und nach Abarbeitung der kleinen Aufgabe das Programm, welches unterbrochen wurde, an der Stelle fortsetzt, an der es vom Interrupt unterbrochen wurde.

0F00	PHA	ACCU
0F01	TXA	UND
0F02	PHA	X - REGISTER RETTEN
0F03	LDA A004	TIMER IM VIA NEU STARTEN
0F06	SED	DEZIMALMODE SETZEN
0F07	LDX #04	X - REGISTER ALS SCHLEIFENZÄHLER
0F09	LDA #01	01 IN ACCU
0F0B	CLC	CARRYBIT LÖSCHEN
0F0C	ADC 0FFB,X	ZÄHLERBYTE UM
0F0F	STA 0FFB,X	1 ERHÖHEN (DEZIMAL)
0F12	CMP 0F2B,X	ENDWERT ERREICHT?
0F15	BNE 0F22	WENN NICHT, DANN SPRUNG
0F17	LDA #00	ENDWERT ERREICHT,
0F19	STA 0FFB,X	ZÄHLERBYTE LÖSCHEN
0F1C	DEX	X DECREMENTIEREN
0F1D	BNE 0F09	WENN X ≠ 0, ERNEUTER DURCHLAUF
0F1F	STX 0FFB	TAGESWERT AUF 00
0F22	CLD	DEZIMALMODE LÖSCHEN
0F23	PLA	ACCU
0F24	TAX	UND X - WERT
0F25	PLA	VOM STAPEL HOLEN
0F26	RTI	RÜCKKEHR VOM INTERRUPT
0F2C	24	
0F2D	60	
0F2E	60	
0F2F	50	

Tabelle der Zählerendwerte

Diese Routine ist auf den ersten Blick nicht leicht zu verstehen. Es wird von der äußerst effektiven indizierten Adressierungsart mit dem X-Register Gebrauch gemacht. Die Zählerbytes, in denen die aktuelle Uhrzeit einprogrammiert werden muß, sind unter Adresse 0FFB – 0FFF angesiedelt, und zwar in folgender Anordnung:

0FFB	-	TAG (VON 00 = SONNTAG BIS 06 = SAMSTAG)
0FFC	-	STD
0FFD	-	MIN
0FFE	-	SEK
0FFF	-	ZÄHLER FÜR 20 ms

Die Zählerendwerte finden sich in der kleinen Tabelle ab Adresse 0F2C. Diese geben Sie am besten mit dem M-Command ein.

Es ist möglich, dieses Programm einer Interruptuhr in einem EPROM unterzubringen. In diesem Fall können Programm und Zählerbytes natürlich nicht auf einer Speicherseite untergebracht werden. Haben Sie derartige Ambitionen, so müssen Sie die entsprechenden Adressen ändern. An sich wäre es sinnvoll, die Zählerbytes in der Zeropage anzusiedeln und mit der indizierten Zeropageadressierung zu bedienen. Da jedoch die Zeropage durch die zahlreichen Programme im AIM schon mehr als reichlich belegt ist, und die Uhrensoftware wirklich unter allen Bedingungen laufen soll, habe ich die Zählerbytes an das Ende des RAM-Bereiches der AIM-Version mit 4K-Bytes gelegt.

Zur Inbetriebnahme des Programms müssen Sie beide Subroutinen einschließlich der kleinen Tabelle eingeben und das Programm ab Adresse 0F30 starten.

Dann können Sie die Uhrzeit mit dem M-Command kontrollieren und verändern.

Obwohl der AIM mit einem Quarz getaktet wird, so gehen die AIM-Uhren doch nicht auf allen Systemen gleich schnell. Sie können Ihr System eventuell ein wenig kalibrieren durch Verändern des Wertes im niederwertigen Byte des Timerlatches von T1 beim Initialisieren (der Wert in Adresse 0F3B).

Im nächsten Heft werden wir den bis hier beschriebenen Weg weiter verfolgen und ein Programm vorstellen, das die aktuelle Uhrzeit auf dem Display anzeigt, aber ohne lästiges Flackern, was mit den bereits besprochenen Monitorsubroutinen nicht ohne weiteres möglich ist. Bis dann wünsche ich Ihnen viel Spaß und Erfolg bei eigenen Versuchen.

Manuel Krüger

TI 58/59 – Besitzer!

Fordern Sie ein kostenloses Probeheft von PPX an.
Die Anwenderzeitschrift für programmierbare Rechner.

In jeder Ausgabe:
– 6 ausführliche Programmlistings
– Tests neuer Rechner
– Preisgünstiges Zubehörangebot
– Viele Tips für's Programmieren
– Umfangreiche Programmbibliothek

48,- DM
nur im Jahresabo!

☎ 0261/37551 · Gespro GmbH · 54 Koblenz 33

Lernen mit AIM, Teil 7

Lernen mit dem AIM, Teil 7 (letzter Teil)

Allgemeines

Der AIM wurde seit dem Frühjahr 1979 weltweit in mehreren tausend Exemplaren verkauft und hat sich vor allem in dem Bereich, für den er konzipiert wurde, nämlich im hardwarenahen Einsatz mit Maschinenprogramm, bewährt. Inzwischen bieten diverse Firmen hard- und softwaremäßige Erweiterungen an: Floppy-Disk, Speicherverweiterungen, Video-Interfaces, EPROM-Programmer, BASIC-Erweiterungen und anderes. Rockwell selbst vertreibt seit kurzem neben den bereits bekannten Programmen BASIC und ASSEMBLER die Programmiersprachen PL/65 und FORTH, letztere vor allem, laut Ankündigung, für Steuerungsanwendungen. Beide passen in die AIM-Steckfassungen. Nicht nur bei Hobbyprogrammierern und als Entwicklungssystem findet er Verwendung, sondern auch bei Firmen, die mit diesem Einplatinencomputer spezielle Entwicklungsaufgaben für Kleinserien lösen. Auf der "electronica 80" konnte man dafür einige interessante Beispiele sehen.

Mit einem Gehäuse, das industriellen Anforderungen genügt und entsprechenden Festwertspeichern in den Stecksockeln des AIM (das EPROM Intel 2716 z.B. läßt sich hierfür mit geringfügigen Änderungen auf der Platine verwenden), dient er Zwecken der Steuerungstechnik, zur Meßwerterfassung- und Überwachung. Diese Geräte haben neben der Grundplatine und eventuell einigen Treiberrouninen zur Ansteuerung von Drucker, Display und Tastatur wenig mit dem AIM gemein, es sind völlig eigenständig arbeitende Geräte.

Jedoch bleibt Rockwell nicht beim erreichten Stand stehen. Auf der "electronica 80" war am Stand der Firma das Nachfolgemodell, der AIM 65/40 zu sehen. Soweit man sehen konnte, verfolgt der neue Rechner das gleiche Konzept wie der alte AIM 65, er sieht ähnlich aus, jedoch um einiges wuchtiger. Dies kommt daher, daß Drucker und Display nun zu 40-stelligen Ausgabeeinheiten wurden. Als CPU dient weiterhin der bewährte 6502. Das Modell auf der Messe war

jedoch vorerst ein Prototyp, es gibt noch technische Schwierigkeiten, und mit einem Verkauf ist wohl nicht vor Mitte 1981 zu rechnen. Der Preis wird, obwohl noch keine präzisen Auskünfte zu erhalten waren, einiges über DM 3000,- liegen.

Seit März 1980, als der erste Teil dieser Serie in ELCOMP erschien, ist dies die siebente und letzte Folge.

Ich selbst arbeite nur noch wenig mit dem AIM, da ich seit einiger Zeit mit einem Entwicklungssystem auf der Basis der CPU 6809 programmiere. Der Microprozessor 6809 stellt eine Weiterentwicklung des 6800 dar, man könnte ihn in gewisser Weise ebenso als Folgemodell des 6502 ansehen. Er beinhaltet viele Möglichkeiten der beiden CPUs, bietet jedoch viel neues, was effektives Programmieren anbelangt. Dies soll jedoch nicht Gegenstand dieser Serie sein.

5.3 Die Ausgaberroutine

In Teil 6 besprachen wir den VIA 6522 im Interruptbetrieb. Ein Anwendungsfall für diese Betriebsart war die Softwareuhr als Subroutine, die dem Bediener auch während der Arbeit im Monitorprogramm, im BASIC oder im Assembler weiterhin zur Verfügung steht. Für das Uhrenprogramm soll Ihnen nun eine Ausgaberroutine vorgestellt werden, die das lästige Flackern der Displays vermeidet, welches entsteht, wenn man die üblichen, dem Anwender zur Verfügung stehenden Monitorroutinen benutzt.

Es gibt eine Einsprungstelle im Monitor, die Subroutine OUTDD1 unter Adresse EF7B, mittels derer man einzelne Anzeigeeinheiten direkt ansprechen kann. Benutzt man diese Routine, so ist die Adresse der entsprechenden Displaystelle im X-Register zu übergeben. Das ASCII-Zeichen, welches ausgegeben werden soll, befindet sich im ACCU. Zu beachten ist dabei, daß das Paritätsbit (bit 7) im ACCU auf 1 gesetzt wird.

Folgendes Hilfsprogramm macht Sie mit der Subroutine OUTDD1 vertraut:

```
0200  ORA #80      PARITÄTSBIT = 1
```

```

0202 JSR EF7B OUTDD1
0205 JMP E182 ZUM MONITOR

```

Mit den Monitorroutinen können Sie den Befehlszähler, ACCU und X-Register entsprechend setzen und dann die Wirkung von OUTDD1 testen.

Im folgenden Programm müssen die BCD-Daten der Zählerbytes unseres Uhrenunterprogramms in den ACCU geladen werden, zu ASCII-Zeichen umgewandelt werden und an die entsprechende Displaystelle mittels Index X gebracht werden. Die Aufteilung eines Acht-Bit-BCD-Wortes geschieht für das höherwertige Digit durch vierfaches rechtsschieben des ACCU, für das niederwertige Digit durch maskieren mit 0F.

Anschließend muß mit 30 oderiert werden, um das entsprechende ASCII-Zeichen zu erhalten. Im folgenden Programm wird mit B0 oderiert. Dies deswegen, um zugleich bis 7 des Zeichens auf 1 zu setzen.

```

#F52 LDX #04 STELLE AUF DISPLAY
#F54 LDY #00 ZEIGER FÜR ZU LADENDEN TABELLENWERT
#F56 LDA #FFB,Y HÖHERWERTIGES
#F59 LSR A DIGIT
#F5A LSR A IN
#F5B LSR A ACCU
#F5C LSR A DURCH SCHIEBEN
#F5D JSR #F77 AUSGABE EINES DIGITS
#F5E LDA #FFB,Y NIEDERWERTIGES DIGIT
#F53 AND #0F IM ACCU MASKIEREN
#F55 JSR #F77 AUSGABE EINES DIGITS
#F60 LDA #A0 LEEERZEICHEN IM ACCU
#F6A JSR #F79 DIREKTAUSGABE OHNE ODERIEREN
#F6D TNY ZEICHT AUF NÄCHSTEN TABELLENWERT
#F6E CPY #05 SIND ALLE WERTE AUSGEGEBEN ?
#F70 BNE #F56 WENN NICHT, DANN SPRUNG
#F72 NOP BLEIBT
#F73 NOP VORERST
#F74 NOP FREI
#F75 BEQ #F52 WENN JA, DANN WIEDER VON VORN
#F77 ORA #B0 ACCU ODERIEREN
#F79 STX #FPA X ZWISCHENSPEICHERN
#F7C JSR EF7B OUTDD1
#F7F LDX #FPA X WIEDER HOLEN
#F82 INX ERHÖHEN; FÜR NÄCHSTE DISPLAYSTELLE
#F83 RTS

```

Gemeinsam mit den beiden Routinen aus der letzten Folge haben wir nun alle wesentlichen Module für das Uhrenprogramm zusammen. Sollte Sie das etwas nervöse Zählen in den letzten beiden Displaystellen stören, so geben Sie in Adresse 0F6E CPY #04 ein. Das Uhrenprogramm läuft sofort nach Starten der Initialisierungsroutine los, wenn Sie den Sprung zum Monitor in Adresse 0F4F durch drei NOP-Befehle ersetzen. Es ist wünschenswert, das Programm durch Drücken etwa der F2-Taste wieder zu verlassen, bzw. durch die F1-Taste erneut anzuspriegen.

Folgende Ergänzung der Routine ermöglicht das Verlassen des Anzeigeprogramms mittels der F2-Taste:

```

#F72 JSR #F84
-----
#F84 FIP
#F85 LDA #BF
#F87 STA A480 DATENREGISTER A (TASTATUR, RIOT 6532)
#F8A BIT A482
#F8D BPL #F91 WENN TASTE GEDRÜCKT, SPRUNG
#F8F PLP ANSONSTEN
#F90 RTS RÜCKSPRUNG
#F91 PLA STACK
#F92 PLA IN ORDNUNG
#F93 PLA BRINGEN
#F94 JMP E182 MONITOR

```

Damit sind wir an das Ende dieses Artikels und zugleich der Serie "LERNEN MIT DEM AIM" gekommen. Alles was an Hard- und Softwarebeispielen in den sieben Folgen von mir dargestellt wurde, sollten erste Hinweise und Beispiele für Ihre eigenen Versuche sein. In diesem Sinne wünsche ich Ihnen viel Freude und Erfolg beim Umsetzen eigener Ideen!

Manuel Krüger, Berlin ■

PIC	PROGRAMMODULE FÜR	PIC-RECHENZENTRUM
	INDIVIDUELLE	PARKSTR. 30
	COMPUTERSOFTWARE	4174 SEVELEN

PIC-GmbH, Europas erstes Softwareversandhaus

Standardprogramme für
CBM und TRS 80 ab 490,— DM

Lager-, Adress-, Karteiverwaltung, Buchführung,
Lohnabrechnung, Baulohn.

Informationsmappe gegen 50 DM Schutzgebühr
Handbücher 100 DM (wird bei Programmkau angeordnet)

Sehr viel Zubehör ab Lager
(u.a. Basic-Editor, Lautsprecher, Notstromversorgung)

Preislisten gegen fertig adressierten A 5 Umschlag und 5 DM
(in Briefmarken oder bar)